400081

National
Semiconductor

Programmable Logic Devices

Databook and Design Guide

1989 Programmable Logic Devices Databook & Design Guide National Semiconductor

Hamilton Avnet
ELECTRONICS
AN AVNET COMPANY
4960 Corporate Drive Suite 135
Huntsville, Al 35805
Telephone # 205-837-7210
MS WATS 1-800-633-2992 AL WATS 1-800-572-2907

A Corporate Dedication to Quality and Reliability

National Semiconductor is an industry leader in the manufacture of high quality, high reliability integrated circuits. We have been the leading proponent of driving down IC defects and extending product lifetimes. From raw material through product design, manufacturing and shipping, our quality and reliability is second to none.

We are proud of our success . . . it sets a standard for others to achieve. Yet, our quest for perfection is ongoing so that you, our customer, can continue to rely on National Semiconductor Corporation to produce high quality products for your design systems.

Charlie Sporck

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Wir fühlen uns zu Qualität und Zuverlässigkeit verpflichtet

National Semiconductor Corporation ist führend bei der Herstellung von integrierten Schaltungen hoher Qualität und hoher Zuverlässigkeit. National Semiconductor war schon immer Vorreiter, wenn es galt, die Zahl von IC Ausfällen zu verringern und die Lebensdauern von Produkten zu verbessern. Vom Rohmaterial über Entwurf und Herstellung bis zur Auslieferung, die Qualität und die Zuverlässigkeit der Produkte von National Semiconductor sind unübertroffen.

Wir sind stolz auf unseren Erfolg, der Standards setzt, die für andere erstrebenswert sind. Auch ihre Ansprüche steigen ständig. Sie als unser Kunde können sich auch weiterhin auf National Semiconductor verlassen.

La Qualité et La Fiabilité:

Une Vocation Commune Chez National Semiconductor Corporation

National Semiconductor Corporation est un des leaders industriels qui fabrique des circuits intégrés d'une très grande qualité et d'une fiabilité exceptionelle. National a été le premier à vouloir faire chuter le nombre de circuits intégrés défectueux et a augmenter la durée de vie des produits. Depuis les matières premières, en passant par la conception du produit sa fabrication et son expédition, partout la qualité et la fiabilité chez National sont sans équivalents.

Nous sommes fiers de notre succès et le standard ainsi défini devrait devenir l'objectif à atteindre par les autres sociétés. Et nous continuons à vouloir faire progresser notre recherche de la perfection; il en résulte que vous, qui êtes notre client, pouvez toujours faire confiance à National Semiconductor Corporation, en produisànt des systèmes d'une très grande qualité standard.

Un Impegno Societario di Qualità e Affidabilità

National Semiconductor Corporation è un'industria al vertice nella costruzione di circuiti integrati di altà qualità ed affidabilità. National è stata il principale promotore per l'abbattimento della difettosità dei circuiti integrati e per l'allungamento della vita dei prodotti. Dal materiale grezzo attraverso tutte le fasi di progettazione, costruzione e spedizione, la qualità e affidabilità National non è seconda a nessuno.

Noi siamo orgogliosi del nostro successo che fissa per gli altri un traguardo da raggiungere. Il nostro desiderio di perfezione è d'altra parte illimitato e pertanto tu, nostro cliente, puoi continuare ad affidarti a National Semiconductor Corporation per la produzione dei tuoi sistemi con elevati livelli di qualità.

Charlie Sporck

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Programmable Logic Devices

DATABOOK AND DESIGN GUIDE

1989 Edition

PART I — DATA BOOK

Product Line Overview 1

Data Sheets 2

PART II — DESIGN GUIDE

Designing with PLDs 3

PLD Design Methodology 4

PLD Design Development Tools 5

Fabrication of Programmable Logic 6

Application Examples 7

Appendices/Physical Dimensions 8

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

- Abuseable™
- Anadig™
- ANS-R-TRAN™
- APPS™
- ASPECT™
- Auto-Chem Deflasher™
- BCP™
- BI-FET™
- BI-FET II™
- BI-LINE™
- BIPLAN™
- BLC™
- BLX™
- Brite-Lite™
- BTL™
- CheckTrack™
- CIM™
- CIMBUS™
- CLASIC™
- Clock✓Chek™
- COMBO™
- COMBO I™
- COMBO II™
- COPS™ microcontrollers
- Datachecker®
- DENSPAK™
- DIB™
- Digitalker®
- DISCERN™
- DISTILL™
- DNR®
- DPVM™
- ELSTAR™
- E-Z-LINK™
- FACT™
- FAIRCAD™
- Fairtech™
- FAST®
- 5-Star Service™
- GENIX™
- GNX™
- HAMR™
- HandiScan™
- HEX 3000™
- HPC™
- I^{3}L®
- ICM™
- INFOCHEX™
- Integral ISE™
- Intelisplay™
- ISE™
- ISE/06™
- ISE/08™
- ISE/16™
- ISE32™
- ISOPLANAR™
- ISOPLANAR-Z™
- KeyScan™
- LMCMOS™
- M^{2}CMOS™
- Macrobus™
- Macrocomponent™
- MAXI-ROM®
- Meat✓Chek™
- MenuMaster™
- Microbus™ data bus
- MICRO-DAC™
- µtalker™
- Microtalker™
- MICROWIRE™
- MICROWIRE/PLUS™
- MOLE™
- MST™
- Naked-8™
- National®
- National Semiconductor®
- National Semiconductor Corp.®
- NAX 800™
- Nitride Plus™
- Nitride Plus Oxide™
- NML™
- NOBUS™
- NSC800™
- NSCISE™
- NSX-16™
- NS-XC-16™
- NTERCOM™
- NURAM™
- OXISS™
- P^{2}CMOS™
- PC Master™
- Perfect Watch™
- Pharma✓Chek™
- PLAN™
- PLANAR™
- Polycraft™
- POSilink™
- POSitalker™
- Power + Control™
- POWERplanar™
- QUAD3000™
- QUIKLOOK™
- RAT™
- RTX16™
- SABR™
- Script✓Chek™
- SCX™
- SERIES/800™
- Series 900™
- Series 3000™
- Series 32000®
- Shelf✓Chek™
- SofChek™
- SPIRE™
- STAR™
- Starlink™
- STARPLEX™
- SuperChip™
- SuperScript™
- SYS32™
- TapePak®
- TDS™
- TeleGate™
- The National Anthem®
- Time✓Chek™
- TINA™
- TLC™
- Trapezoidal™
- TRI-CODE™
- TRI-POLY™
- TRI-SAFE™
- TRI-STATE®
- TURBOTRANSCEIVER™
- VIP™
- VR32™
- WATCHDOG™
- XMOS™
- XPU™
- Z STAR™
- 883B/RETS™
- 883S/RETS™

E^{2}CMOS™ is a trademark and GAL® and UltraMOS® are registered trademarks of Lattice Semiconductor
PAL® and PALASM™ are trademarks of and used under license from Monolithic Memories, Inc.
IBM® is a registered trademark of International Business Machines Corp.
VAX™ is a trademark of Digital Equipment Corportation
ABEL™ is a trademark of Data I/O
CUPL™ is a trademark of Logical Devices

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000
TWX (910) 339-9240

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry or specifications.

Introduction

HOW TO USE THIS BOOK

This Data Book and Design Guide is intended for use by system designers and manufacturers who wish to take advantage of the opportunities created with National's Programmable Logic Devices (PLDs). The book goes beyond a collection of available data sheets by adding selected application examples and background information on design techniques and benefits of programmable logic products.

The book is laid out to permit users of varying familiarity with PLDs to find the information they need. It should be used as a reference work at all levels and as an introductory tutorial for those areas which require review. While PLDs provide a level of simplicity, flexibility and compactness beyond that offered by other off-the-shelf parts, their optimal use requires the designer to be familiar with design methodologies, development tools and design testing as well as the details of the products themselves.

Experienced users will find reference information in Part I (Data Book). Part I contains two sections which will aid the designer during actual system design and development process. Section 1 provides an overview of the entire programmable logic product line available from National to aid in device selection. Section 2 presents complete data sheets for all listed devices.

For the user new to PLD design, it is recommended that the introductory Section 3 is read first, followed by careful study of Section 4 on design methodology, with reference to PLD Tools in Section 5 and PLD Fabrication (technology and quality) in Section 6 as required. Following this, a review of available products in Section 1 and examination of a few data sheets of interest in Section 2 will provide background for a study of concrete design examples enumerated in Section 7.

National
Semiconductor

Product Status Definitions

Definition of Terms

Data Sheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This data sheet contains preliminary data, and supplementary data will be published at a later date. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

National Semiconductor Corporation reserves the right to make changes without further notice to any products herein to improve reliability, function or design. National does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

Table of Contents

Alphanumeric Index xi

PART I—DATA BOOK

Section 1 Product Line Overview

1.1 Selection Tables 1-3
1.2 Ordering Information 1-7
1.3 Block Diagrams 1-8

Section 2 Data Sheets

20-Pin Small PAL Family 2-3
20-Pin Medium PAL Family 2-23
24-Pin Small PAL Family 2-47
24-Pin Exclusive-OR PAL Family 2-61
24-Pin Medium PAL Family 2-77
24-Pin Polarity PAL Family 2-95
PAL16RA8 2-109
PAL20RA10 2-119
GAL16V8 Generic Array Logic 2-129
GAL20V8 Generic Array Logic 2-145
GAL16V8A Generic Array Logic (Preliminary) 2-163
GAL20V8A Generic Array Logic (Preliminary) 2-167
GAL16Z8 In-System Re-programmable (isp) Generic Array Logic (Advance Information) 2-171
GAL39V18 Generic Array Logic (Advance Information) 2-175
ECL PAL10/10016P8 2-183
ECL Registered/Latched PAL Family (Preliminary) 2-193
ECL PAL10/10016P4A 2-207
ECL PAL10/10012C4A (Preliminary) 2-211
ECL PAL10/10016RM4A, PAL10/10016LM4A (Preliminary) 2-217

PART II—DESIGN GUIDE

Section 3 Designing with Programmable Logic

3.1 Background to PLDs 3-3
  PROM devices
  FPLA devices
  PAL devices
  GAL devices
3.2 Design Advantages 3-3
  Simplified system design
  Increased functional density
3.3 Manufacturing Advantages 3-5
  Cost of inventory
  Cost of ownership
  Cost of upgrades and modifications
  Higher reliability
3.4 Alternative Methodologies 3-6
  Standard LSI
  Standard SSI/MSI
  Full custom
  Semi-custom ASICs

Table of Contents (Continued)

Section 3 Designing with Programmable Logic (Continued)
3.5 PLD Architecture Overview 3-6
Logic array structure
Registered outputs
Exclusive-OR gates
Programmable output polarity
Advanced product features
3.6 Programmable Logic from National 3-11
TTL PROMs
Lateral fuse TTL PAL products
Vertical fuse TTL PAL products
ECL PAL products
E^{2}CMOS GAL products
Pre-programmed and mask-programmed products
3.7 Design Development Tools 3-12
Programming hardware
Development software
NSC PLAN software
Other assemblers/compilers
Section 4 Programmable Logic Design Methodology
4.1 Design Development Process 4-3
4.2 Logic Design 4-3
Defining the problem
Designing the logic
Deriving Boolean equations
Logic minimization
4.3 Design Implementation 4-6
Selecting a device
Design/equation entry
Compilation—creating the JEDEC file
Programming hardware preparation
Device programming
4.4 Logic Verification 4-8
Pattern verfication
Test vector generation
Design simulation
Device functional testing
Section 5 PLD Design Development Tools
5.1 Selecting Development Tools 5-3
Programming hardware selection
Selecting the development software
5.2 PLD Programmers 5-3
Types of programmers
Programmer certification
Data I/O Model 29
Data I/O Unisite
Logical Devices ALLPRO
Logic Devices PALPRO-2X
Stag Microsystems PPZ
Stag Microsystems ZL30A

Table of Contents (Continued)

Section 5 PLD Design Development Tools (Continued)

5.3 National PLAN Software 5-5
  - Features of PLAN software
  - PLAN software design entry
  - Design compilation
  - Back translation
  - PAL-to-GAL map conversion

5.4 Other Assemblers/Compilers 5-8
  - PALASM software
  - ABEL software
  - CUPL software

5.5 Pre-Assembler Software Tools 5-8

Section 6 Fabrication of Programmable Logic

6.1 PLD Technologies 6-3
  - Standard, "Series-A" and "Series-B" TTL PAL devices
  - ECL PAL Devices
  - "Series-D" TTL PAL Devices with vertical fuses
  - E^2CMOS GAL Devices

6.2 Quality and Factory Testing 6-5
  - Product reliability
  - Test circuitry
  - Normal PLD factory test
  - Pre-programmed PLD factory test
  - NML factory test
  - Customer handling and test

Section 7 Application Examples

7.1 Basic Gates (PAL12H6) 7-3
7.2 Basic Flip-Flops (GAL16V8) 7-7
7.3 Memory-Mapped I/O Address Decoder (PAL16L2) 7-12
7.4 Quad 4-to-1 Multiplexer (GAL20V8) 7-15
7.5 Dual 8-to-1 Multiplexer (PAL20L2) 7-19
7.6 16-to-1 Multiplexer (PAL20C1) 7-22
7.7 4-Bit Counter with 2 Parallel-Load Ports (PAL16R4) 7-25
7.8 7-Bit Counter with Parallel-Load (GAL20V8) 7-28
7.9 8-Bit Counter with Parallel-Load (PAL20X8) 7-32
7.10 10-Bit Up/Down Counter (PAL20X10) 7-33
7.11 8-Bit Cascadable Shift Register (PAL20X8) 7-37
7.12 8-Bit Equality Comparator (PAL16C1) 7-40
7.13 8-Bit Barrel Shifter (GAL20V8) 7-41
7.14 8-Input Priority Encoder (GAL16V8) 7-45
7.15 16-Input Priority Encoder (GAL20V8) 7-49
7.16 Hexadecimal 7-Segment Display Encoder (PAL16L8) 7-53
7.17 Dual-Port RAM Controller (GAL16V8) 7-58
7.18 8086 CPU Board Random Control Logic (PAL12H6) 7-67
7.19 DP84312 DRAM Controller Interface for NS32032 (PAL16R6) 7-69
7.20 DP84322 DRAM Controller Interface for 68000 CPU (PAL16R4) 7-77
7.21 DP84332 DRAM Controller Interface for 8086/8088 (PAL16R8) 7-94
7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032 7-104

Table of Contents (Continued)

Section 8 Appendices/Physical Dimensions

Appendix A—Boolean Logic Review 8-3
  A.1 Basic Operators and Theorems 8-3
  A.2 Derivation of a Boolean Expression 8-4
  A.3 Minimization 8-6
  A.4 K-Map Method 8-7
  A.5 Sequential Circuit Elements 8-11
  A.6 State Machine Fundamentals 8-12
  A.7 Avoiding Logic Hazards 8-14
Appendix B—Theory of PLD Testing 8-17
  B.1 Testing Methods 8-17
  B.2 Test Generation Techniques 8-18
  B.3 Response Evaluation Techniques 8-28
Appendix C—Physical Dimensions 8-33

Bookshelf

Distributors

Alpha-Numeric Index

GAL16V8-20L 20-Pin Generic Array Logic 2-129
GAL16V8-25L 20-Pin Generic Array Logic 2-129
GAL16V8-25Q 20-Pin Generic Array Logic 2-129
GAL16V8-30L 20-Pin Generic Array Logic 2-129
GAL16V8-30Q 20-Pin Generic Array Logic 2-129
GAL16V8A-10 20-Pin Generic Array Logic (Preliminary) 2-163
GAL16V8A-12 20-Pin Generic Array Logic (Preliminary) 2-163
GAL16V8A-15 20-Pin Generic Array Logic (Preliminary) 2-163
GAL16V8A-20 20-Pin Generic Array Logic (Preliminary) 2-163
GAL16Z8 In-System Re-programmable (isp) Generic Array Logic (Advance Information) 2-171
GAL20V8-20L 24-Pin Generic Array Logic 2-145
GAL20V8-25L 24-Pin Generic Array Logic 2-145
GAL20V8-25Q 24-Pin Generic Array Logic 2-145
GAL20V8-30L 24-Pin Generic Array Logic 2-145
GAL20V8-30Q 24-Pin Generic Array Logic 2-145
GAL20V8A-10 24-Pin Generic Array Logic (Preliminary) 2-167
GAL20V8A-12 24-Pin Generic Array Logic (Preliminary) 2-167
GAL20V8A-15 24-Pin Generic Array Logic (Preliminary) 2-167
GAL20V8A-20 24-Pin Generic Array Logic (Preliminary) 2-167
GAL39V18 Generic Array Logic (Advance Information) 2-175
PAL10012C4A ECL 4 ns PAL (Preliminary) 2-211
PAL10016LD4 ECL Latched PAL (Preliminary) 2-193
PAL10016LD8 ECL Latched PAL (Preliminary) 2-193
PAL10016LM4A ECL Latched PAL (Preliminary) 2-217
PAL10016P4A ECL 4 ns PAL 2-207
PAL10016P8 ECL PAL 2-183
PAL10016RC4 ECL Registered PAL (Preliminary) 2-193
PAL10016RC8 ECL Registered PAL (Preliminary) 2-193
PAL10016RD4 ECL Registered PAL (Preliminary) 2-193
PAL10016RD8 ECL Registered PAL (Preliminary) 2-193
PAL10016RM4A ECL Registered PAL (Preliminary) 2-217
PAL1012C4A ECL 4 ns PAL (Preliminary) 2-211
PAL1016LD4 ECL Latched PAL (Preliminary) 2-193
PAL1016LD8 ECL Latched PAL (Preliminary) 2-193
PAL1016LM4A ECL Latched PAL (Preliminary) 2-217
PAL1016P4A ECL 4 ns PAL 2-207
PAL1016P8 ECL PAL 2-183
PAL1016RC4 ECL Registered PAL (Preliminary) 2-193
PAL1016RC8 ECL Registered PAL (Preliminary) 2-193
PAL1016RD4 ECL Registered PAL (Preliminary) 2-193
PAL1016RD8 ECL Registered PAL (Preliminary) 2-193
PAL1016RM4A ECL Registered PAL (Preliminary) 2-217
PAL10H8 20-Pin Small PAL 2-3
PAL10H8A 20-Pin Small PAL 2-3
PAL10H8A2 20-Pin Small PAL 2-3
PAL10L8 20-Pin Small PAL 2-3
PAL10L8A 20-Pin Small PAL 2-3
PAL10L8A2 20-Pin Small PAL 2-3
PAL12H6 20-Pin Small PAL 2-3
PAL12H6A 20-Pin Small PAL 2-3
PAL12H6A2 20-Pin Small PAL 2-3
PAL12L6 20-Pin Small PAL 2-3

Alpha-Numeric Index (Continued)

PAL12L6A 20-Pin Small PAL 2-3
PAL12L6A2 20-Pin Small PAL 2-3
PAL12L10 24-Pin Small PAL 2-47
PAL12L10A 24-Pin Small PAL 2-47
PAL14H4 20-Pin Small PAL 2-3
PAL14H4A 20-Pin Small PAL 2-3
PAL14H4A2 20-Pin Small PAL 2-3
PAL14L4 20-Pin Small PAL 2-3
PAL14L4A 20-Pin Small PAL 2-3
PAL14L4A2 20-Pin Small PAL 2-3
PAL14L8 24-Pin Small PAL 2-47
PAL14L8A 24-Pin Small PAL 2-47
PAL16C1 20-Pin Small PAL 2-3
PAL16C1A 20-Pin Small PAL 2-3
PAL16C1A2 20-Pin Small PAL 2-3
PAL16H2 20-Pin Small PAL 2-3
PAL16H2A 20-Pin Small PAL 2-3
PAL16H2A2 20-Pin Small PAL 2-3
PAL16L2 20-Pin Small PAL 2-3
PAL16L2A 20-Pin Small PAL 2-3
PAL16L2A2 20-Pin Small PAL 2-3
PAL16L6 24-Pin Small PAL 2-47
PAL16L6A 24-Pin Small PAL 2-47
PAL16L8 20-Pin Medium PAL 2-23
PAL16L8A 20-Pin Medium PAL 2-23
PAL16L8A2 20-Pin Medium PAL 2-23
PAL16L8B 20-Pin Medium PAL 2-23
PAL16L8B2 20-Pin Medium PAL 2-23
PAL16L8D 20-Pin Medium PAL 2-23
PAL16R4 20-Pin Medium PAL 2-23
PAL16R4A 20-Pin Medium PAL 2-23
PAL16R4A2 20-Pin Medium PAL 2-23
PAL16R4B 20-Pin Medium PAL 2-23
PAL16R4B2 20-Pin Medium PAL 2-23
PAL16R4D 20-Pin Medium PAL 2-23
PAL16R6 20-Pin Medium PAL 2-23
PAL16R6A 20-Pin Medium PAL 2-23
PAL16R6A2 20-Pin Medium PAL 2-23
PAL16R6B 20-Pin Medium PAL 2-23
PAL16R6B2 20-Pin Medium PAL 2-23
PAL16R6D 20-Pin Medium PAL 2-23
PAL16R8 20-Pin Medium PAL 2-23
PAL16R8A 20-Pin Medium PAL 2-23
PAL16R8A2 20-Pin Medium PAL 2-23
PAL16R8B 20-Pin Medium PAL 2-23
PAL16R8B2 20-Pin Medium PAL 2-23
PAL16R8D 20-Pin Medium PAL 2-23
PAL16RA8 2-109
PAL18L4 24-Pin Small PAL 2-47
PAL18L4A 24-Pin Small PAL 2-47
PAL20C1 24-Pin Small PAL 2-47
PAL20C1A 24-Pin Small PAL 2-47

Alpha-Numeric Index (Continued)

PAL20L2 24-Pin Small PAL 2-47
PAL20L2A 24-Pin Small PAL 2-47
PAL20L8A 24-Pin Medium PAL 2-77
PAL20L8B 24-Pin Medium PAL 2-77
PAL20L8D 24-Pin Medium PAL (Preliminary) 2-77
PAL20L10 24-Pin XOR PAL 2-61
PAL20L10A 24-Pin XOR PAL 2-61
PAL20P8B 24-Pin Polarity PAL 2-95
PAL20R4A 24-Pin Medium PAL 2-77
PAL20R4B 24-Pin Medium PAL 2-77
PAL20R4D 24-Pin Medium PAL (Preliminary) 2-77
PAL20R6A 24-Pin Medium PAL 2-77
PAL20R6B 24-Pin Medium PAL 2-77
PAL20R6D 24-Pin Medium PAL (Preliminary) 2-77
PAL20R8A 24-Pin Medium PAL 2-77
PAL20R8B 24-Pin Medium PAL 2-77
PAL20R8D 24-Pin Medium PAL (Preliminary) 2-77
PAL20RA10 2-119
PAL20RP4B 24-Pin Polarity PAL 2-95
PAL20RP6B 24-Pin Polarity PAL 2-95
PAL20RP8B 24-Pin Polarity PAL 2-95
PAL20X4 24-Pin XOR PAL 2-61
PAL20X4A 24-Pin XOR PAL 2-61
PAL20X8 24-Pin XOR PAL 2-61
PAL20X8A 24-Pin XOR PAL 2-61
PAL20X10 24-Pin XOR PAL 2-61
PAL20X10A 24-Pin XOR PAL 2-61

PART I
DATA BOOK

Product Line Overview 1

Data Sheets 2

Section 1

Product Line Overview

Section 1 Contents

1.1 Selection Tables 1-3
1.2 Ordering Information 1-7
1.3 Block Diagrams 1-8

National Semiconductor

Section 1
Product Line Overview

Section 1 is provided as a reference guide to National Semiconductor's programmable logic product line. The product selection tables in Section 1.1 and the collection of product block diagrams in Section 1.2 can be used to compare the features and architectures of the various PLD families and to aid in the selection of the appropriate devices which best suit the designer's application.

1.1 Selection Tables

The product selection tables which follow are organized by technology group (TTL, E^{2}CMOS and ECL), then by "family", then by "series" within each family. The term "family" refers to a set of one or more "device types" which are variations on the same basic architecture. The term "device type" refers to a specific device architecture (regardless of performance characteristics). The term "series" refers to a particular speed/power version in which the devices of a PLD family are available. Thus as technology advancements provide for improved speed/power performance, a new series is added to selected product families.

TABLE 1-1. Programmable Logic Product Selection Guide

Family and Series	Part Numbers	t_{PD} (max) (Note 1)	I_{CC} (max)	Outputs		Page
				Combinatorial	Registered	
TTL						
20-Pin Small PAL® (Standard Speed)	PAL10H8	35	90	8	—	2-3
	PAL10L8	35	90	8	—	2-3
	PAL12H6	35	90	6	—	2-3
	PAL12L6	35	90	6	—	2-3
	PAL14H4	35	90	4	—	2-3
	PAL14L4	35	90	4	—	2-3
	PAL16H2	35	90	2	—	2-3
	PAL16L2	35	90	2	—	2-3
	PAL16C1	35	90	1	—	2-3
20-Pin Small PAL Series-A	PAL10H8A	25	90	8	—	2-3
	PAL10L8A	25	90	8	—	2-3
	PAL12H6A	25	90	6	—	2-3
	PAL12L6A	25	90	6	—	2-3
	PAL14H4A	25	90	4	—	2-3
	PAL14L4A	25	90	4	—	2-3
	PAL16H2A	25	90	2	—	2-3
	PAL16L2A	25	90	2	—	2-3
	PAL16C1A	30	90	1	—	2-3
20-Pin Small PAL Series-A2	PAL10H8A2	35	45	8	—	2-3
	PAL10L8A2	35	45	8	—	2-3
	PAL12H6A2	35	45	6	—	2-3
	PAL12L6A2	35	45	6	—	2-3
	PAL14H4A2	35	45	4	—	2-3
	PAL14L4A2	35	45	4	—	2-3
	PAL16H2A2	35	45	2	—	2-3
	PAL16L2A2	35	45	2	—	2-3
	PAL16C1A2	40	45	1	—	2-3
20-Pin Medium PAL (Standard)	PAL16L8	35	180	8	—	2-23
	PAL16R4	35	180	4	4	2-23
	PAL16R6	35	180	2	6	2-23
	PAL16R8	35	180	—	8	2-23

Note 1: Maximum t_{PD} for combinatorial outputs (commercial operating range). Denotes characteristic speed of family where product has all registered outputs.

TABLE 1-1. Programmable Logic Product Selection Guide (Continued)

Family and Series	Part Numbers	t_{PD} (max) (Note 1)	I_{CC} (max)	Outputs		Page
				Combinatorial	Registered	
TTL (Continued)						
20-Pin	PAL16L8A	25	180	8	—	2-23
Medium	PAL16R4A	25	180	4	4	2-23
PAL	PAL16R6A	25	180	2	6	2-23
Series-A	PAL16R8A	25	180	—	8	2-23
20-Pin	PAL16L8A2	35	90	8	—	2-23
Medium	PAL16R4A2	35	90	4	4	2-23
PAL	PAL16R6A2	35	90	2	6	2-23
Series-A2	PAL16R8A2	35	90	—	8	2-23
20-Pin	PAL16L8B	15	180	8	—	2-23
Medium	PAL16R4B	15	180	4	4	2-23
PAL	PAL16R6B	15	180	2	6	2-23
Series-B	PAL16R8B	15	180	—	8	2-23
20-Pin	PAL16L8B2	25	90	8	—	2-23
Medium	PAL16R4B2	25	100	4	4	2-23
PAL	PAL16R6B2	25	100	2	6	2-23
Series-B2	PAL16R8B2	25	100	—	8	2-23
20-Pin	PAL16L8D	10	180	8	—	2-23
Medium	PAL16R4D	10	180	4	4	2-23
PAL	PAL16R6D	10	180	2	6	2-23
Series-D	PAL16R8D	10	180	—	8	2-23
24-Pin	PAL12L10	40	100	10	—	2-47
Small	PAL14L8	40	100	8	—	2-47
PAL	PAL16L6	40	100	6	—	2-47
(Standard	PAL18L4	40	100	4	—	2-47
Speed)	PAL20L2	40	100	2	—	2-47
	PAL20C1	40	100	1	—	2-47
24-Pin	PAL12L10A	25	100	10	—	2-47
Small	PAL14L8A	25	100	8	—	2-47
PAL	PAL16L6A	25	100	6	—	2-47
Series-A	PAL18L4A	25	100	4	—	2-47
	PAL20L2A	25	100	2	—	2-47
	PAL20C1A	30	100	1	—	2-47
24-Pin	PAL20L10	50	165	10	—	2-61
XOR	PAL20X4	50	180	6	4	2-61
PAL	PAL20X8	50	180	2	8	2-61
(Standard)	PAL20X10	50	180	—	10	2-61
24-Pin	PAL20L10A	30	165	10	—	2-61
XOR	PAL20X4A	30	180	6	4	2-61
PAL	PAL20X8A	30	180	2	8	2-61
Series-A	PAL20X10A	30	180	—	10	2-61
24-Pin	PAL20L8A	25	210	8	—	2-77
Medium	PAL20R4A	25	210	4	4	2-77
PAL	PAL20R6A	25	210	2	6	2-77
Series-A	PAL20R8A	25	210	—	8	2-77
24-Pin	PAL20L8B	15	210	8	—	2-77
Medium	PAL20R4B	15	210	4	4	2-77
PAL	PAL20R6B	15	210	2	6	2-77
Series-B	PAL20R8B	15	210	—	8	2-77

Note 1: Maximum t_{PD} for combinatorial outputs (commercial operating range). Denotes characteristic speed of family where product has all registered outputs.

TABLE 1-1. Programmable Logic Product Selection Guide (Continued)

Family and Series	Part Numbers	t_{PD} (max) (Note 1)	I_{CC} (max)	Outputs		Page
				Combinatorial	Registered	
TTL (Continued)						
24-Pin Medium PAL Series-D	PAL20L8D	10	210	8	—	2-77
	PAL20R4D	10	210	4	4	2-77
	PAL20R6D	10	210	2	6	2-77
	PAL20R8D	10	210	—	8	2-77
24-Pin Polarity PAL Series-B	PAL20P8B	15	210	8	—	2-95
	PAL20RP4B	15	210	4	4	2-95
	PAL20RP6B	15	210	2	6	2-95
	PAL20RP8B	15	210	—	8	2-95
Registered Asynchronous	PAL16RA8	30	170	—	8	2-109
	PAL20RA10	30	200	—	10	2-119
E^{2}CMOS						
20-Pin Generic Array Logic	GAL16V8-20L	20	90	—	8	2-129
	GAL16V8-25Q	25	45	—	8	2-129
	GAL16V8-25L	25	90	—	8	2-129
	GAL16V8-30Q	30	45	—	8	2-129
	GAL16V8-30L	30	90	—	8	2-129
20-Pin Generic Array Logic Series-A	GAL16V8A-10	10	115	—	8	2-163
	GAL16V8A-12	12	115	—	8	2-163
	GAL16V8A-15	15	115	—	8	2-163
	GAL16V8A-20	20	115	—	8	2-163
24-Pin Generic Array Logic	GAL20V8-20L	20	90	—	8	2-145
	GAL20V8-25Q	25	45	—	8	2-145
	GAL20V8-25L	25	90	—	8	2-145
	GAL20V8-30Q	30	45	—	8	2-145
	GAL20V8-30L	30	90	—	8	2-145
24-Pin Generic Array Logic Series-A	GAL20V8A-10	10	115	—	8	2-167
	GAL20V8A-12	12	115	—	8	2-167
	GAL20V8A-15	15	115	—	8	2-167
	GAL20V8A-20	20	115	—	8	2-167

Note 1: Maximum t_{PD} for combinatorial outputs (commercial operating range). Denotes characteristic speed of family where product has all registered outputs.

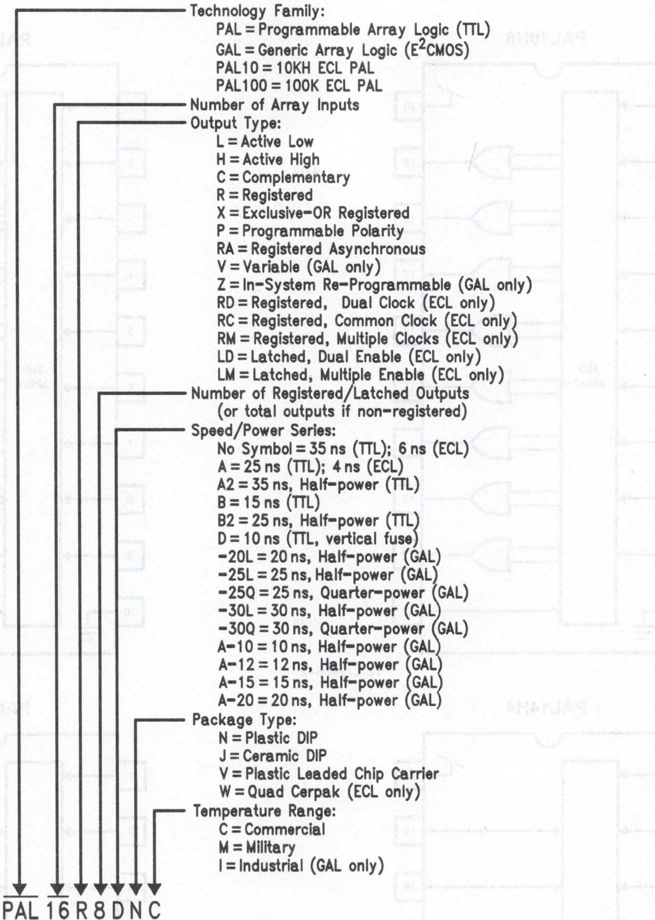
TABLE 1-1. Programmable Logic Product Selection Guide (Continued)

Family and Series	Part Numbers	t_{PD} (max) (Note 1)	I_{CC} (max)	Outputs		Page
				Combinatorial	Registered	
ECL						
Combinatorial	PAL1016P8	6	−240	8	—	2-183
	PAL10016P8	6	−240	8	—	2-183
Registered	PAL1016RD8	6	−280	—	8	2-193
	PAL10016RD8	6	−280	—	8	2-193
	PAL1016RC8	6	−280	—	8	2-193
	PAL10016RC8	6	−280	—	8	2-193
	PAL1016RD4	6	−260	4	4	2-193
	PAL10016RD4	6	−260	4	4	2-193
	PAL1016RC4	6	−260	4	4	2-193
	PAL10016RC4	6	−260	4	4	2-193
Latched	PAL1016LD8	6	−260	—	8	2-193
	PAL10016LD8	6	−260	—	8	2-193
	PAL1016LD4	6	−260	4	4	2-193
	PAL10016LD4	6	−260	4	4	2-193
Combinatorial Series-A	PAL1016P4A	4	−220	4	—	2-207
	PAL10016P4A	4	−220	4	—	2-207
	PAL1012C4A	4	−220	4	—	2-211
	PAL10012C4A	4	−220	4	—	2-211
Registered Series-A	PAL1016RM4A	4	−220	—	4	2-217
	PAL10016RM4A	4	−220	—	4	2-217
Latched Series-A	PAL1016LM4A	4	−220	—	4	2-217
	PAL10016LM4A	4	−220	—	4	2-217

Note 1: Maximum t_{PD} for combinatorial outputs (commercial operating range). Denotes characteristic speed of family where product has all registered outputs.

1.2 Ordering Information

The ordering information diagram below defines the product-number nomenclature used throughout National's programmable logic product line. This nomenclature is based on that used by the original industry-standard PAL products, and are therefore very similar to the product numbers used by other PLD manufacturers. Refer to the corresponding "Ordering Information" diagrams in the individual product datasheets to determine the valid combinations of attributes describing actual PLD products.



Technology Family:
- PAL = Programmable Array Logic (TTL)
- GAL = Generic Array Logic (E^2CMOS)
- PAL10 = 10KH ECL PAL
- PAL100 = 100K ECL PAL

Number of Array Inputs

Output Type:
- L = Active Low
- H = Active High
- C = Complementary
- R = Registered
- X = Exclusive-OR Registered
- P = Programmable Polarity
- RA = Registered Asynchronous
- V = Variable (GAL only)
- Z = In-System Re-Programmable (GAL only)
- RD = Registered, Dual Clock (ECL only)
- RC = Registered, Common Clock (ECL only)
- RM = Registered, Multiple Clocks (ECL only)
- LD = Latched, Dual Enable (ECL only)
- LM = Latched, Multiple Enable (ECL only)

Number of Registered/Latched Outputs
(or total outputs if non-registered)

Speed/Power Series:
- No Symbol = 35 ns (TTL); 6 ns (ECL)
- A = 25 ns (TTL); 4 ns (ECL)
- A2 = 35 ns, Half-power (TTL)
- B = 15 ns (TTL)
- B2 = 25 ns, Half-power (TTL)
- D = 10 ns (TTL, vertical fuse)
- -20L = 20 ns, Half-power (GAL)
- -25L = 25 ns, Half-power (GAL)
- -25Q = 25 ns, Quarter-power (GAL)
- -30L = 30 ns, Half-power (GAL)
- -30Q = 30 ns, Quarter-power (GAL)
- A-10 = 10 ns, Half-power (GAL)
- A-12 = 12 ns, Half-power (GAL)
- A-15 = 15 ns, Half-power (GAL)
- A-20 = 20 ns, Half-power (GAL)

Package Type:
- N = Plastic DIP
- J = Ceramic DIP
- V = Plastic Leaded Chip Carrier
- W = Quad Cerpak (ECL only)

Temperature Range:
- C = Commercial
- M = Military
- I = Industrial (GAL only)

TL/L/9986-1

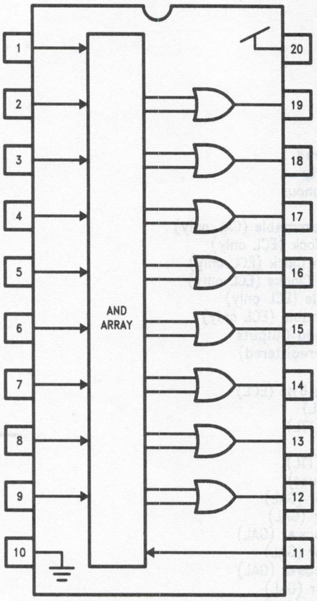
FIGURE 1-1. PLD Part-Number Nomenclature and Ordering Information

1.3 Block Diagrams

The following collection of block diagrams provides a graphic representation of all product architectures ("device types") in National's programmable logic product line. The block diagrams may be used in conjunction with the selection tables in Section 1.1 to identify the most appropriate PLD products to suit the system's application requirements.

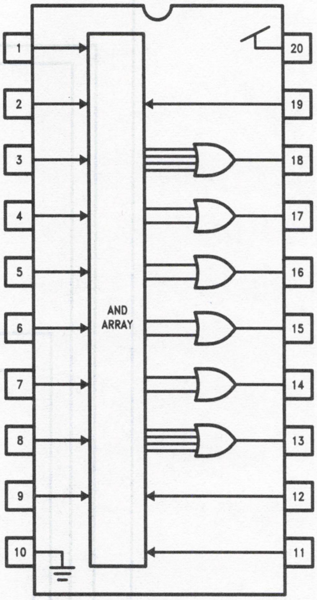
20-Pin Small PAL Family

PAL10H8


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

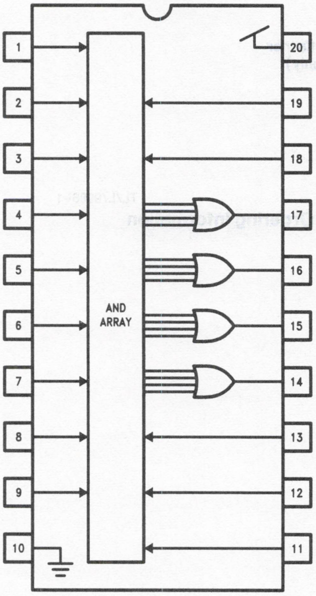
TL/L/9986-2

PAL12H6


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

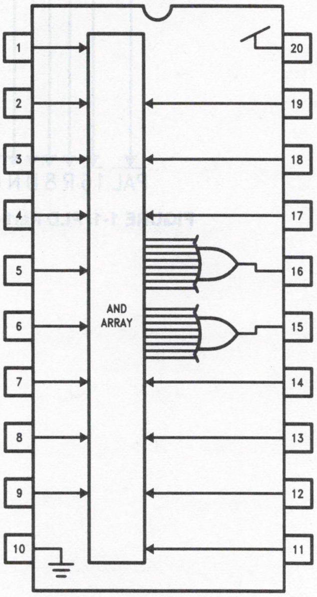
TL/L/9986-3

PAL14H4


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

TL/L/9986-4

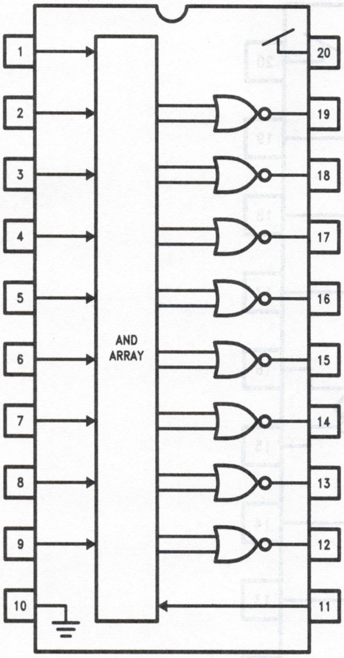
PAL16H2


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

TL/L/9986-5

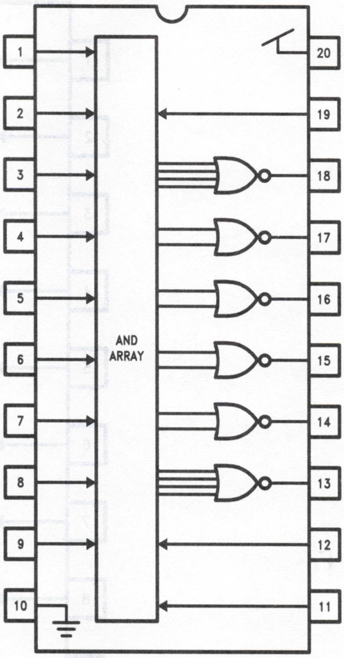
20-Pin Small PAL Family (Continued)

PAL10L8


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

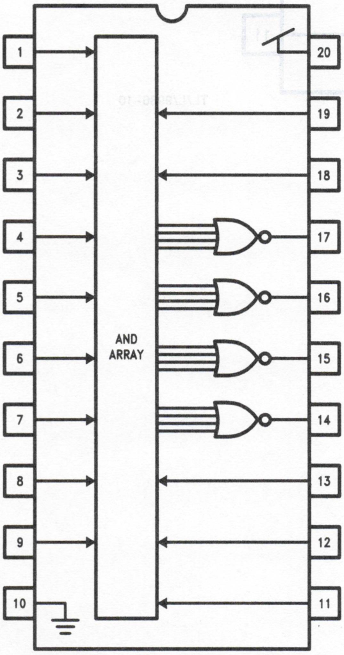
TL/L/9986-6

PAL12L6


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

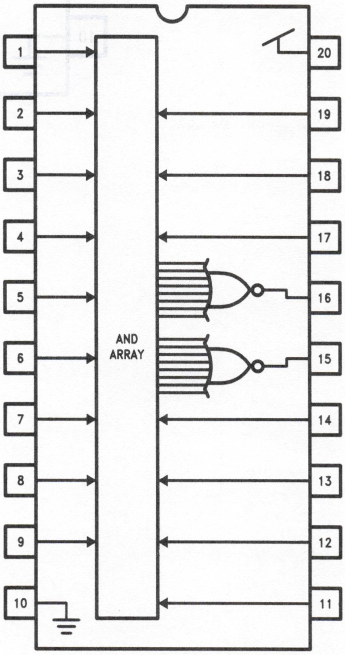
TL/L/9986-7

PAL14L4


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

TL/L/9986-8

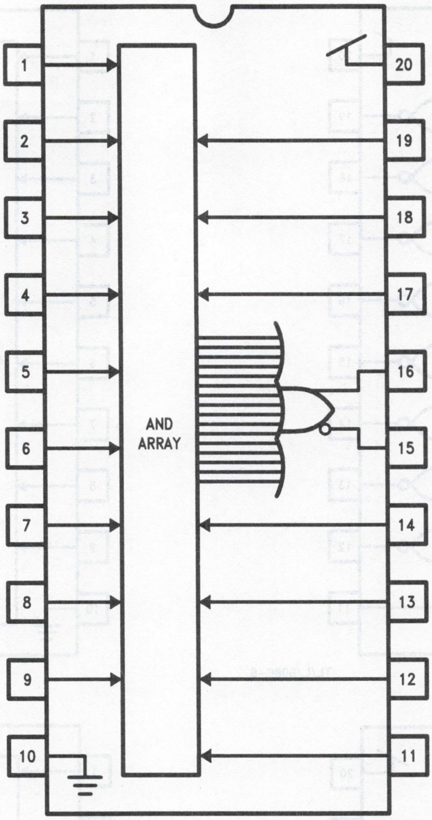
PAL16L2


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

TL/L/9986-9

20-Pin Small PAL Family (Continued)

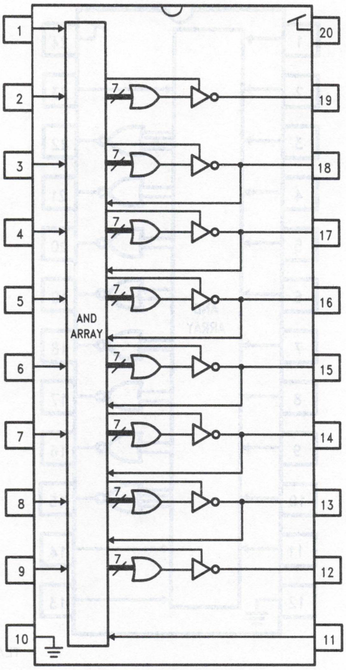
PAL16C1


1
2
3
4
5
6
7
8
9
10
AND ARRAY
20
19
18
17
16
15
14
13
12
11

TL/L/9986-10

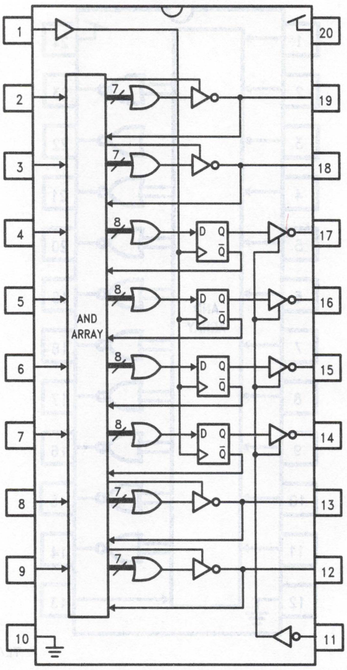
20-Pin Medium PAL Family

PAL16L8


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
AND ARRAY
7

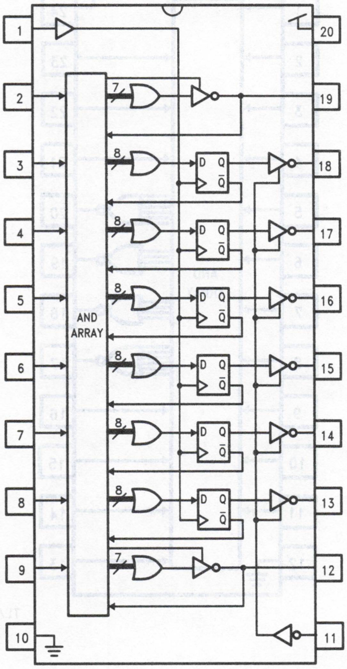
TL/L/9986-11

PAL16R4


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
AND ARRAY
7
8
D
Q
Q̄

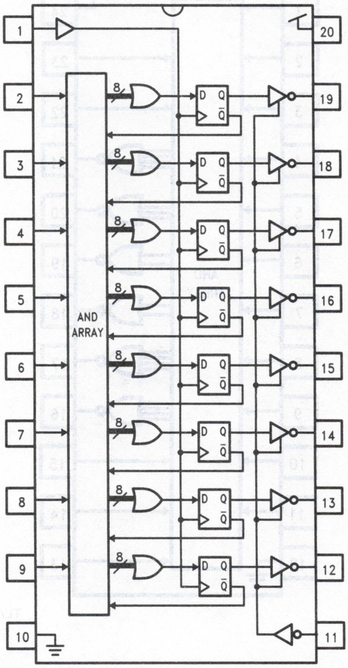
TL/L/9986-12

PAL16R6


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
AND ARRAY
7
8
D
Q
Q̄

TL/L/9986-13

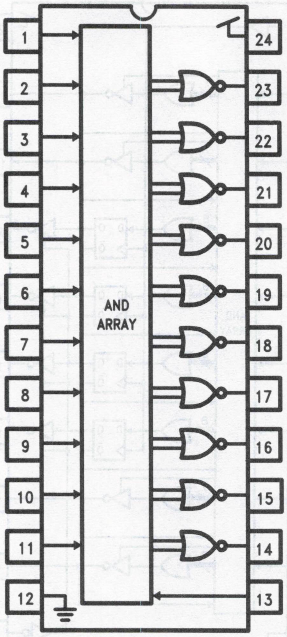
PAL16R8


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
AND ARRAY
8
D
Q
Q̄

TL/L/9986-14

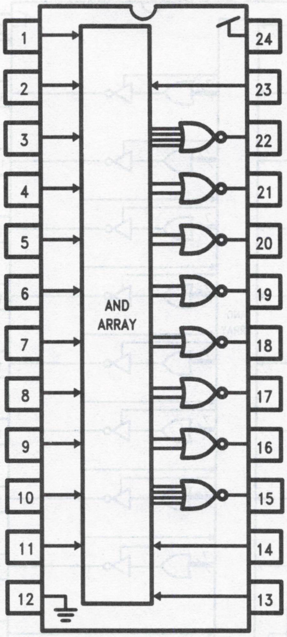
24-Pin Small PAL Family

PAL12L10


1
2
3
4
5
6
7
8
9
10
11
12
AND
ARRAY
24
23
22
21
20
19
18
17
16
15
14
13

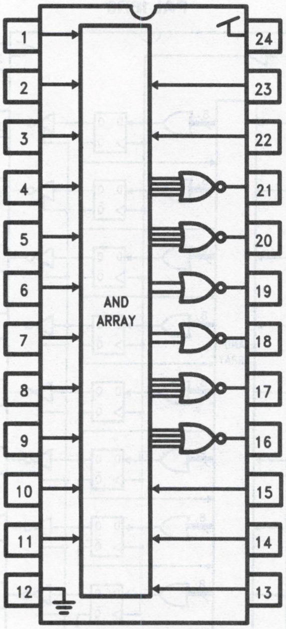
TL/L/9986-15

PAL14L8


1
2
3
4
5
6
7
8
9
10
11
12
AND
ARRAY
24
23
22
21
20
19
18
17
16
15
14
13

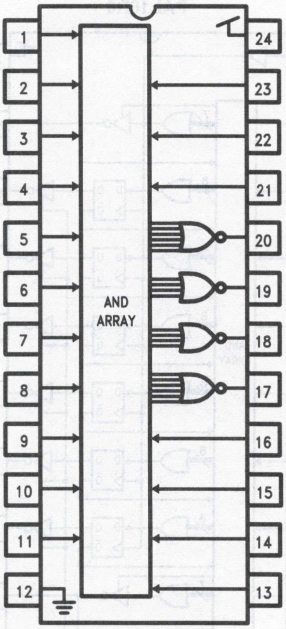
TL/L/9986-16

PAL16L6


1
2
3
4
5
6
7
8
9
10
11
12
AND
ARRAY
24
23
22
21
20
19
18
17
16
15
14
13

TL/L/9986-17

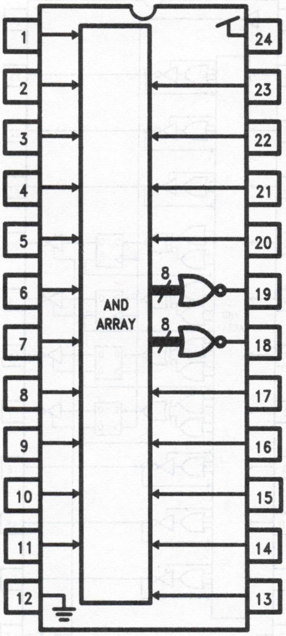
PAL18L4


1
2
3
4
5
6
7
8
9
10
11
12
AND
ARRAY
24
23
22
21
20
19
18
17
16
15
14
13

TL/L/9986-18

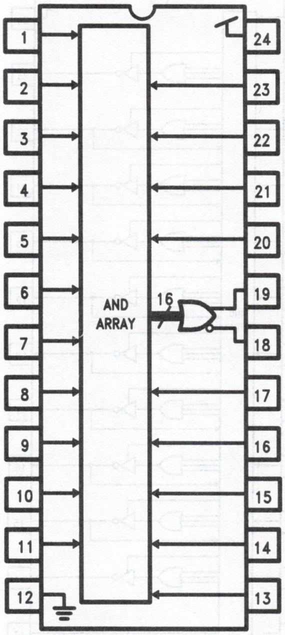
24-Pin Small PAL Family (Continued)

PAL20L2


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND
ARRAY
8
8

TL/L/9986-19

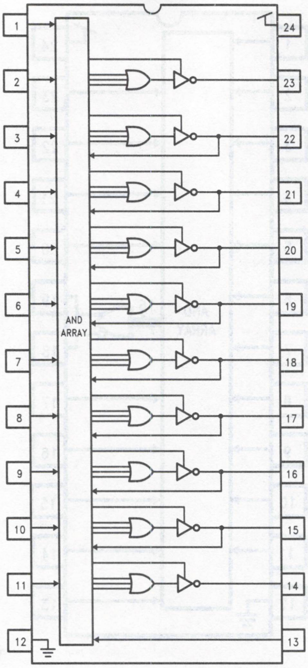
PAL20C1


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND
ARRAY
16

TL/L/9986-20

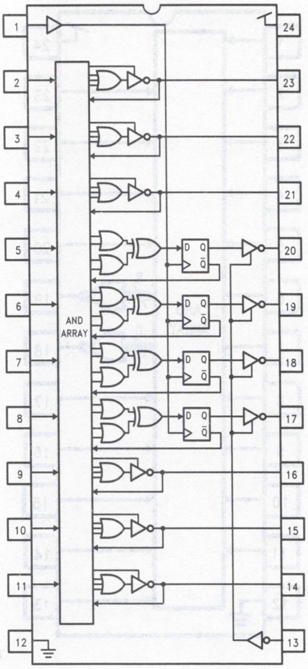
24-Pin Exclusive-OR PAL Family

PAL20L10


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY

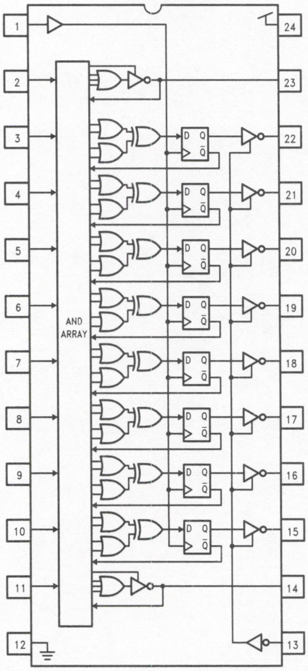
TL/L/9986-21

PAL20X4


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
D Q
Q̄

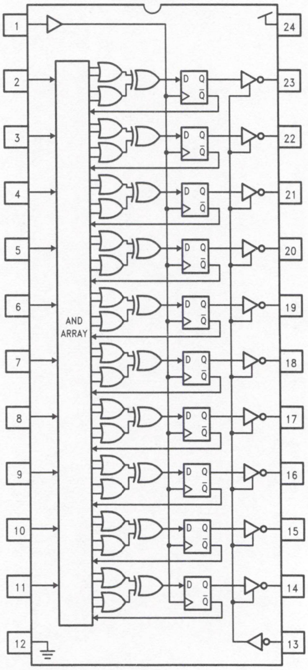
TL/L/9986-22

PAL20X8


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
D Q
Q̄

TL/L/9986-23

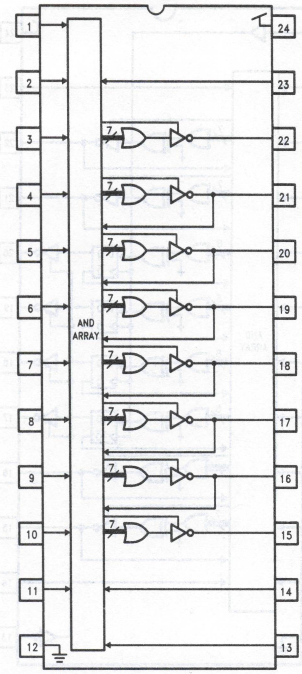
PAL20X10


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
D Q
Q̄

TL/L/9986-24

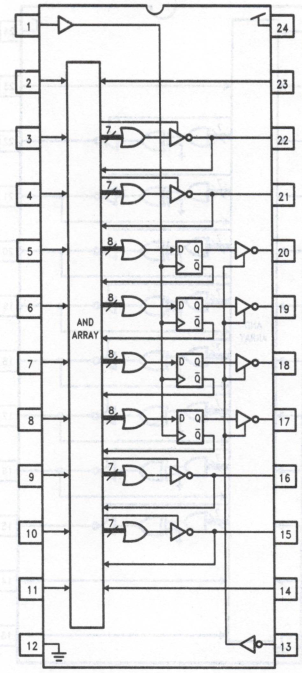
24-Pin Medium PAL Family

PAL20L8


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
7

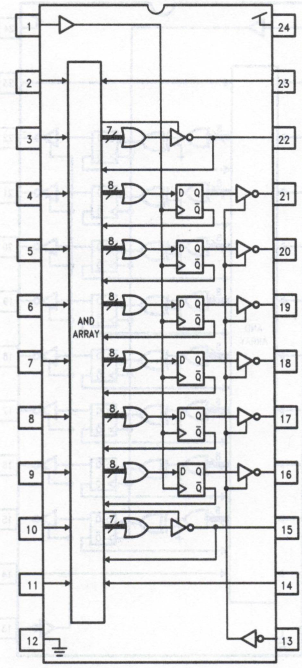
TL/L/9986-25

PAL20R4


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
7
8
D
Q
Q̄

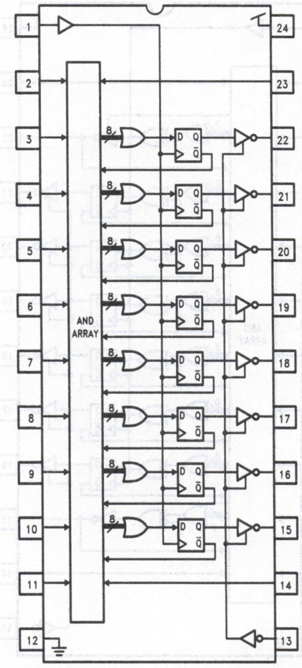
TL/L/9986-26

PAL20R6


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
7
8
D
Q
Q̄

TL/L/9986-27

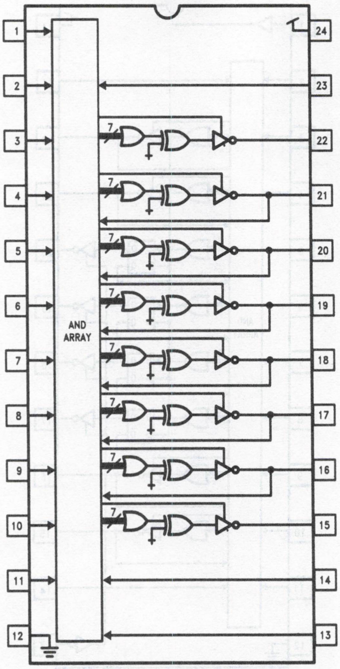
PAL20R8


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
8
D
Q
Q̄

TL/L/9986-28

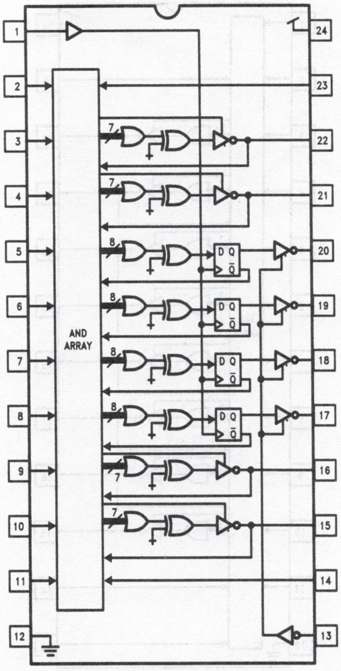
24-Pin Polarity PAL Family

PAL20P8


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
AND ARRAY
7

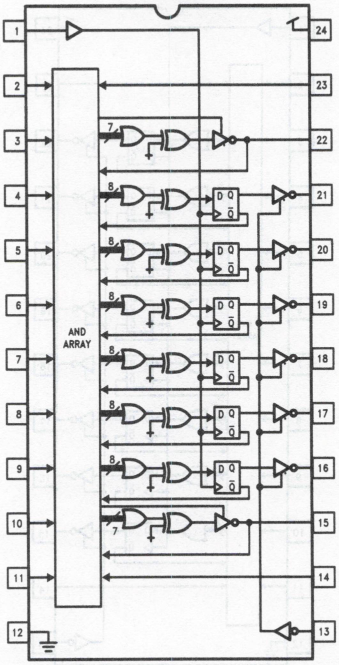
TL/L/9986-29

PAL20RP4


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
AND ARRAY
7
8
D Q
Q̄

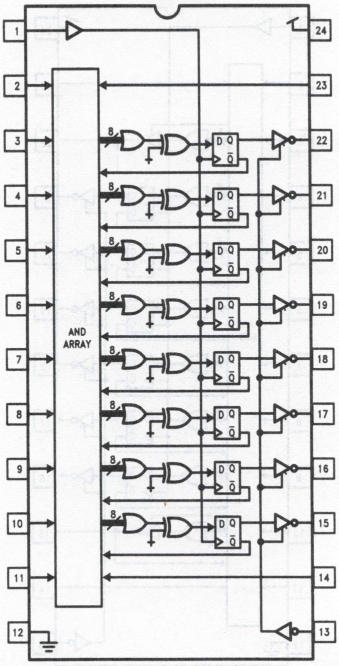
TL/L/9986-30

PAL20RP6


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
AND ARRAY
7
8
D Q
Q̄

TL/L/9986-31

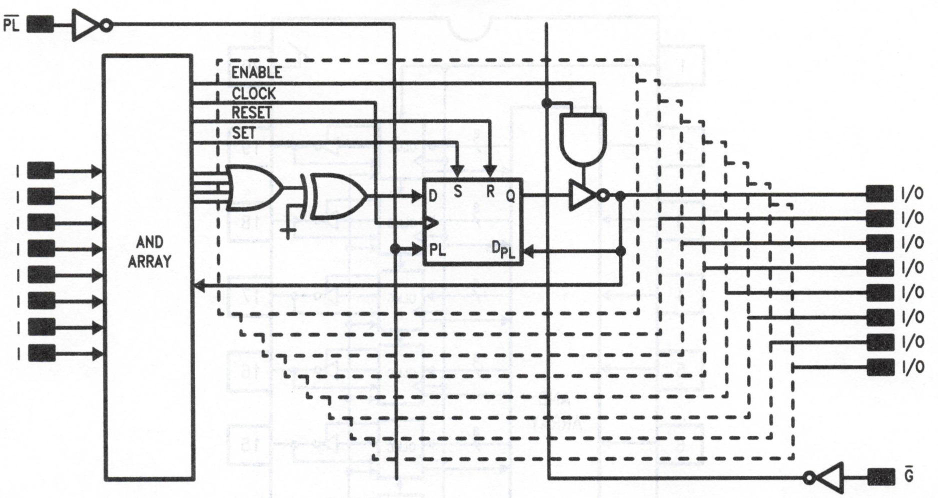
PAL20RP8


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
AND ARRAY
8
D Q
Q̄

TL/L/9986-32

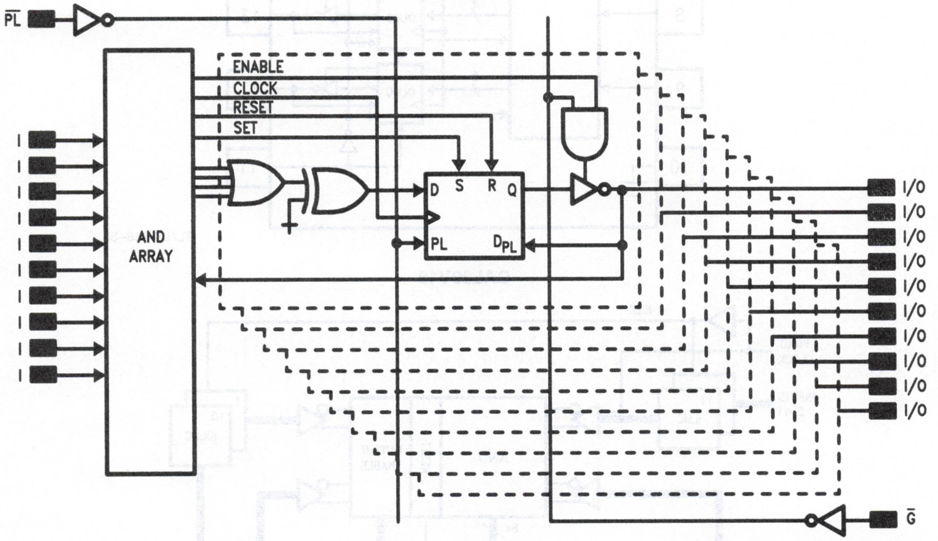
Registered Asynchronous PAL Devices

PAL16RA8


PL
ENABLE
CLOCK
RESET
SET
AND ARRAY
D
S
R
Q
PL
D_PL
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
G

TL/L/9986-33

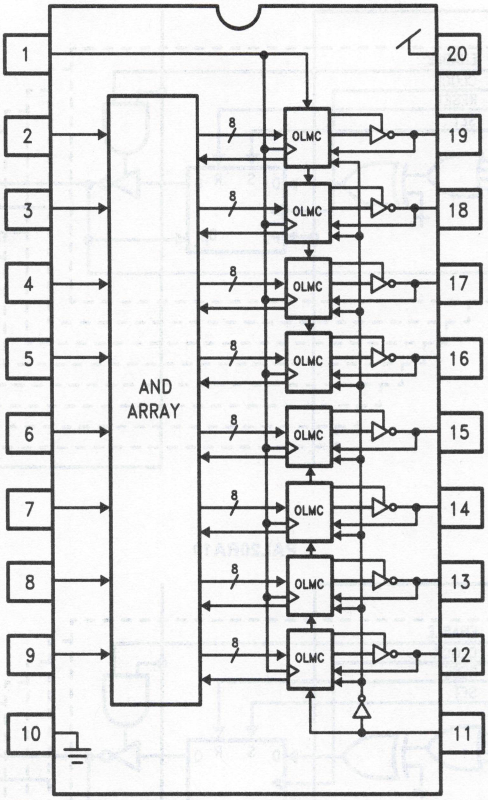
PAL20RA10


PL
ENABLE
CLOCK
RESET
SET
AND ARRAY
D
S
R
Q
PL
D_PL
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
G

TL/L/9986-34

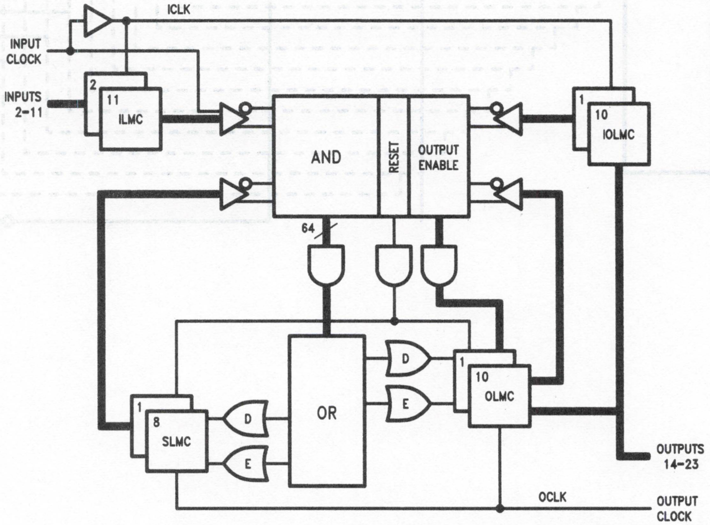
Generic Array Logic Family

GAL16V8


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
AND
ARRAY
8
OLMC

TL/L/9986-35

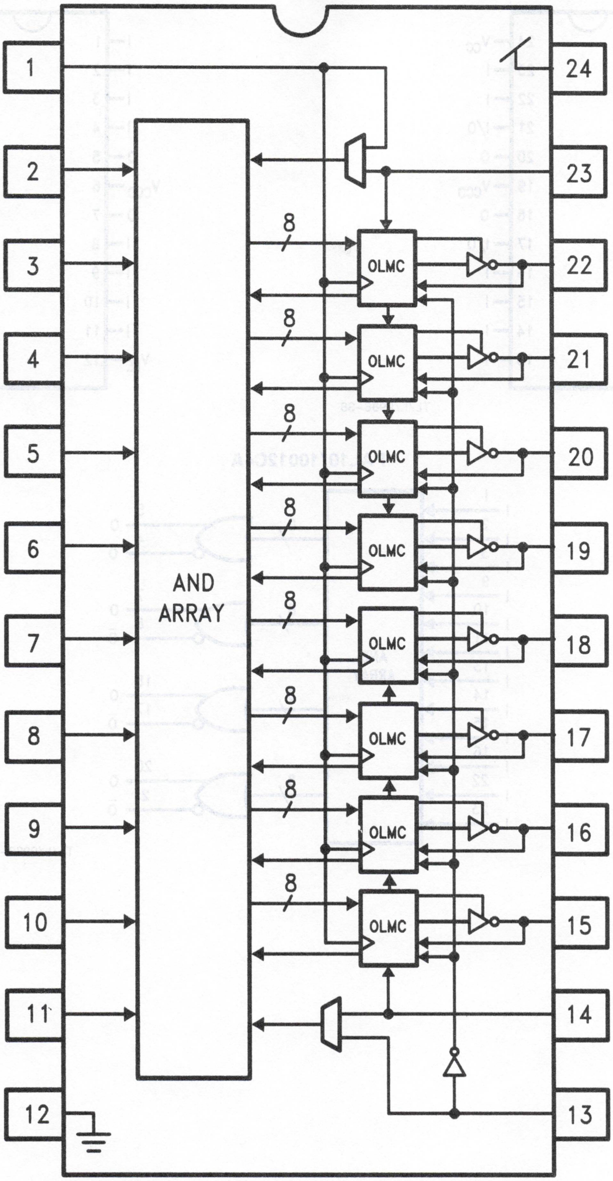
GAL39V18


ICLK
INPUT
CLOCK
INPUTS
2–11
2
11
ILMC
AND
RESET
OUTPUT
ENABLE
1
10
IOLMC
64
OR
D
E
1
10
OLMC
1
8
SLMC
D
E
OUTPUTS
14–23
OCLK
OUTPUT
CLOCK

TL/L/9986-37

Generic Array Logic Family (Continued)

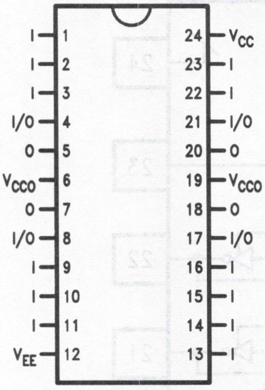
GAL20V8


1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
AND ARRAY
8
OLMC

TL/L/9986-36

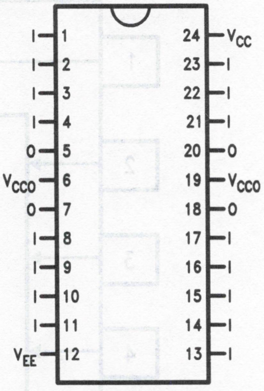
ECL Combinatorial PAL Family

PAL 10/10016P8


I 1
I 2
I 3
I/O 4
O 5
VCCO 6
O 7
I/O 8
I 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 I/O
20 O
19 VCCO
18 O
17 I/O
16 I
15 I
14 I
13 I

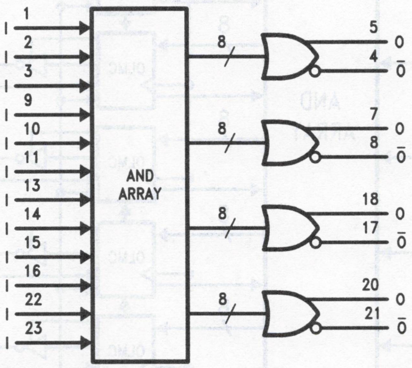
TL/L/9986-38

PAL10/10016P4A


I 1
I 2
I 3
I 4
O 5
VCCO 6
O 7
I 8
I 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 I
20 O
19 VCCO
18 O
17 I
16 I
15 I
14 I
13 I

TL/L/9986-39

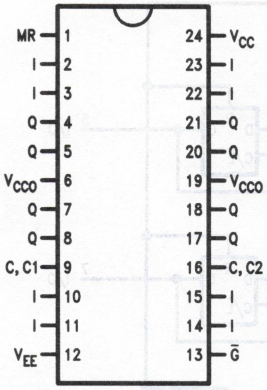
PAL10/10012C4A


I 1
I 2
I 3
I 9
I 10
I 11
I 13
I 14
I 15
I 16
I 22
I 23
AND ARRAY
8
8
8
8
5 O
4 Ō
7 O
8 Ō
18 O
17 Ō
20 O
21 Ō

TL/L/9986-40

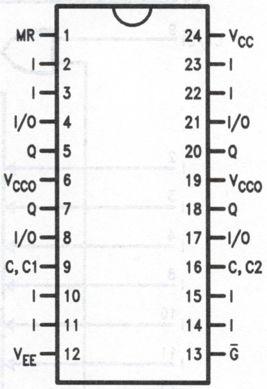
ECL Registered/Latched PAL Family

PAL10/10016RD8
PAL10/10016RC8


MR 1
I 2
I 3
Q 4
Q 5
VCCO 6
Q 7
Q 8
C, C1 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 Q
20 Q
19 VCCO
18 Q
17 Q
16 C, C2
15 I
14 I
13 G̅

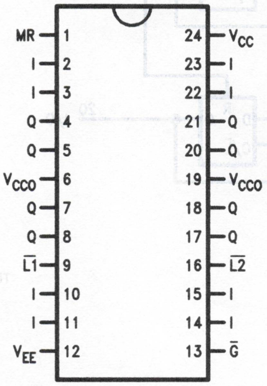
TL/L/9986-41

PAL10/10016RD4
PAL10/10016RC4


MR 1
I 2
I 3
I/O 4
Q 5
VCCO 6
Q 7
I/O 8
C, C1 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 I/O
20 Q
19 VCCO
18 Q
17 I/O
16 C, C2
15 I
14 I
13 G̅

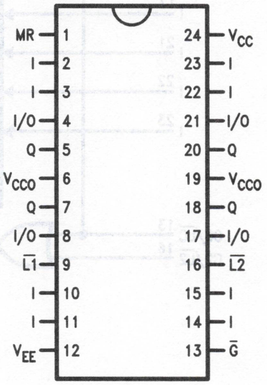
TL/L/9986-42

PAL10/10016LD8


MR 1
I 2
I 3
Q 4
Q 5
VCCO 6
Q 7
Q 8
L̅1̅ 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 Q
20 Q
19 VCCO
18 Q
17 Q
16 L̅2̅
15 I
14 I
13 G̅

TL/L/9986-43

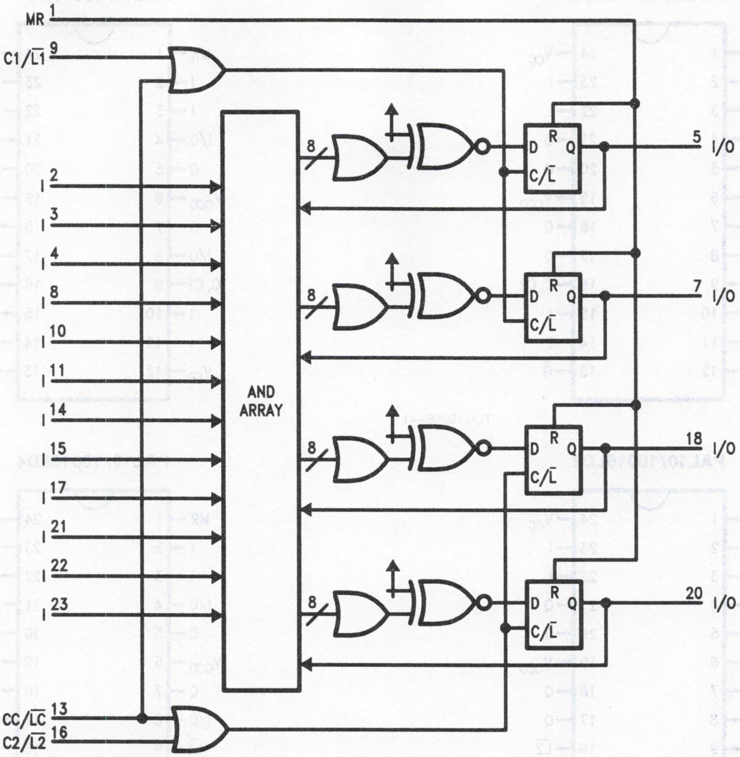
PAL10/10016LD4


MR 1
I 2
I 3
I/O 4
Q 5
VCCO 6
Q 7
I/O 8
L̅1̅ 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 I/O
20 Q
19 VCCO
18 Q
17 I/O
16 L̅2̅
15 I
14 I
13 G̅

TL/L/9986-44

ECL Registered/Latched PAL Family (Continued)

PAL10/10016RM4A
PAL10/10016LM4A


MR 1
C1/$\overline{L1}$ 9
I 2
I 3
I 4
I 8
I 10
I 11
I 14
I 15
I 17
I 21
I 22
I 23
CC/$\overline{LC}$ 13
C2/$\overline{L2}$ 16
AND ARRAY
8
8
8
8
D
R
Q
C/$\overline{L}$
5 I/O
7 I/O
18 I/O
20 I/O

TL/L/9986-47

Section 2
Data Sheets

Section 2 Contents

20-Pin Small PAL Family 2-3
20-Pin Medium PAL Family 2-23
24-Pin Small PAL Family 2-47
24-Pin Exclusive-OR PAL Family 2-61
24-Pin Medium PAL Family 2-77
24-Pin Polarity PAL Family 2-95
PAL16RA8 2-109
PAL20RA10 2-119
GAL16V8 Generic Array Logic 2-129
GAL20V8 Generic Array Logic 2-145
GAL16V8A Generic Array Logic (Preliminary) 2-163
GAL20V8A Generic Array Logic (Preliminary) 2-167
GAL16Z8 In-System Re-programmable (isp) Generic Array Logic (Advance Information) 2-171
GAL39V18 Generic Array Logic (Advance Information) 2-175
ECL PAL10/10016P8 2-183
ECL Registered/Latched PAL Family (Preliminary) 2-193
ECL PAL10/10016P4A 2-207
ECL PAL10/10012C4A (Preliminary) 2-211
ECL PAL10/10016RM4A, PAL10/10016LM4A (Preliminary) 2-217

National Semiconductor

Programmable Array Logic (PAL®) 20-Pin Small PAL Family

General Description

The 20-pin Small PAL family contains nine popular PAL architectures. The devices in the Small PAL family draw only 90 mA maximum supply current for standard power versions, and as little as 45 mA for Series A2 as compared to 180 mA in the 20-pin Medium PAL devices. These devices offer speeds as fast as 25 ns maximum propagation delay. National Semiconductor's Schottky TTL process with titanium tungsten fusible links provides high-speed user-programmable replacements for conventional SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming fusible links to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

The Small PAL logic array has between 10 and 16 complementary input pairs and up to 8 combinatorial outputs generated by a single programmable AND-gate array with fixed OR-gate connections. The Small PAL family offers a variety of input/output combinations as shown in the Device Types table below. Security fuses can be programmed to prevent direct copying of proprietary logic patterns.

Features

- As fast as 25 ns maximum propagation delay
- User-programmable replacement for TTL logic
- Large variety of JEDEC-compatible programming equipment available
- Fully supported by National PLAN™ development software
- Security fuse prevents direct copying of logic patterns

Device Types

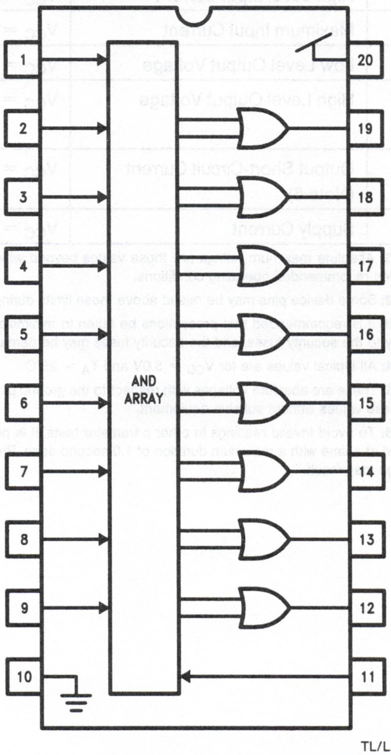
Device Type	Dedicated Inputs	Combinatorial Outputs
PAL10H8/PAL10L8	10	8
PAL12H6/PAL12L6	12	6
PAL14H4/PAL14L4	14	4
PAL16H2/PAL16L2	16	2
PAL16C1	16	1 Pair

Speed/Power Versions

Series	Example	Commercial		Military	
		t_{PD}	I_{CC}	t_{PD}	I_{CC}
Standard	PAL10H8	35 ns	90 mA	45 ns	90 mA
A	PAL10H8A	25 ns*	90 mA	30 ns*	90 mA
A2	PAL10H8A2	35 ns*	45 mA	45 ns	45 mA

*Except PAL16C1A t_{PD} = 30 ns Commercial, 35 ns Military.
PAL16C1A2 t_{PD} = 40 ns Commercial.

Block Diagram—PAL10H8



TL/L/9995-1

Standard Series (PAL10H8, PAL12H6, PAL14H4, PAL16H2, PAL10L8, PAL12L6, PAL14L4, PAL16L2, PAL16C1)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5 to +7.0V
Input Voltage (Notes 2 and 3)	−1.5 to +5.5V
Off-State Output Voltage (Note 2)	−1.5 to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	Military			Commercial			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature	−55			0		75	°C
T_C	Operating Care Temperature			125				°C

Electrical Characteristics Over Recommended Operating Conditions (Note 4)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 5)						0.8	V
V_{IH}	High Level Input Voltage (Note 5)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 8 mA			0.3	0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OS}	Output Short-Circuit Current (Note 6)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				55	90	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: All typical values are for V_{CC} = 5.0V and T_A = 25°C.

Note 5: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

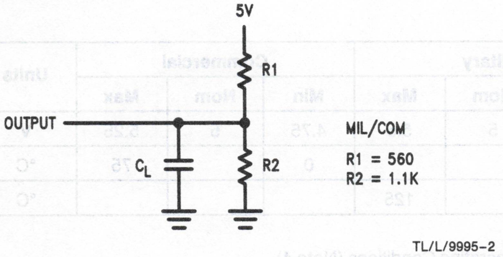
Note 6: To avoid invalid readings in other parameter tests, it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 second each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Standard Series (PAL10H8, PAL12H6, PAL14H4, PAL16H2, PAL10L8, PAL12L6, PAL14L4, PAL16L2, PAL16C1)
(Continued)

Switching Characteristics Over Recommended Operating Conditions

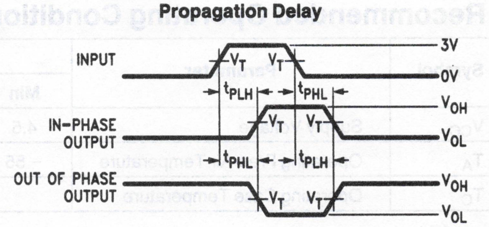
Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input to Output	$C_L = 50$ pF		25	45		25	35	ns

Test Load


5V
R1
OUTPUT
C_L
R2
MIL/COM
R1 = 560
R2 = 1.1K

TL/L/9995-2

Test Waveform


Propagation Delay
INPUT
3V
0V
V_T
V_T
t_{PLH}
t_{PHL}
IN-PHASE OUTPUT
V_{OH}
V_{OL}
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT
V_{OH}
V_{OL}

TL/L/9995-3

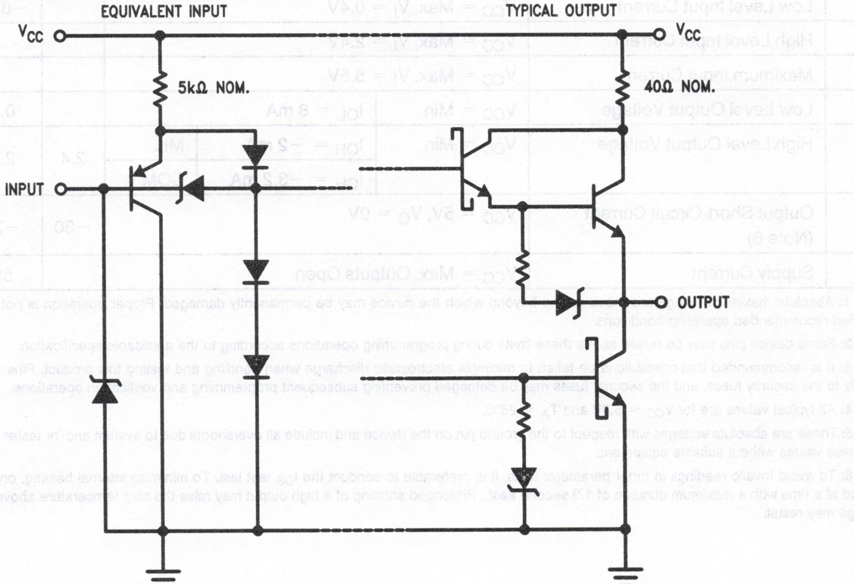
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Schematic of Inputs and Outputs


EQUIVALENT INPUT
TYPICAL OUTPUT
V_{CC}
V_{CC}
5kΩ NOM.
40Ω NOM.
INPUT
OUTPUT

TL/L/9995-4

Series A (PAL10H8A, PAL12H6A, PAL14H4A, PAL16H2A, PAL10L8A, PAL12L6A, PAL14L4A, PAL16L2A, PAL16C1A)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5 to +7.0V
Input Voltage (Notes 2 and 3)	−1.5 to +5.5V
Off-State Output Voltage (Note 2)	−1.5 to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	Military			Commercial			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature	−55			0		75	°C
T_C	Operating Case Temperature			125				°C

Electrical Characteristics Over Recommended Operating Conditions (Note 4)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 5)						0.8	V
V_{IH}	High Level Input Voltage (Note 5)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 8 mA			0.3	0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OS}	Output Short-Circuit Current (Note 6)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				55	90	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: All typical values are for V_{CC} = 5.0V and T_A = 25°C.

Note 5: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

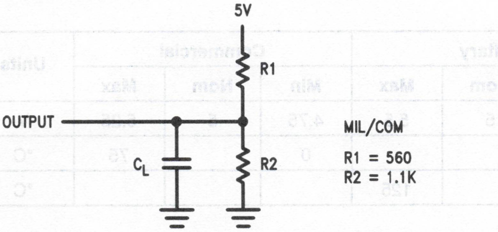
Note 6: To avoid invalid readings in other parameter tests, it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 second each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Series A (PAL10H8A, PAL12H6A, PAL14H4A, PAL16H2A, PAL10L8A, PAL12L6A, PAL14L4A, PAL16L2A, PAL16C1A) (Continued)

Switching Characteristics Over Recommended Operating Conditions

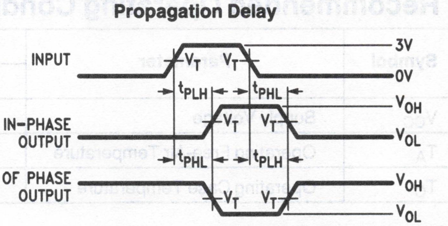
Symbol	Parameter	Test Conditions		Military			Commercial			Units
				Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input to Output	C_L = 50 pF	Except 16C1A		15	30		15	25	ns
			16C1A			35			30	ns

Test Load


5V
R1
OUTPUT
C_L
R2
MIL/COM
R1 = 560
R2 = 1.1K

TL/L/9995-2

Test Waveform


Propagation Delay
INPUT
3V
0V
V_T
V_T
t_{PLH}
t_{PHL}
V_{OH}
V_{OL}
IN-PHASE OUTPUT
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT
V_{OH}
V_{OL}

TL/L/9995-3

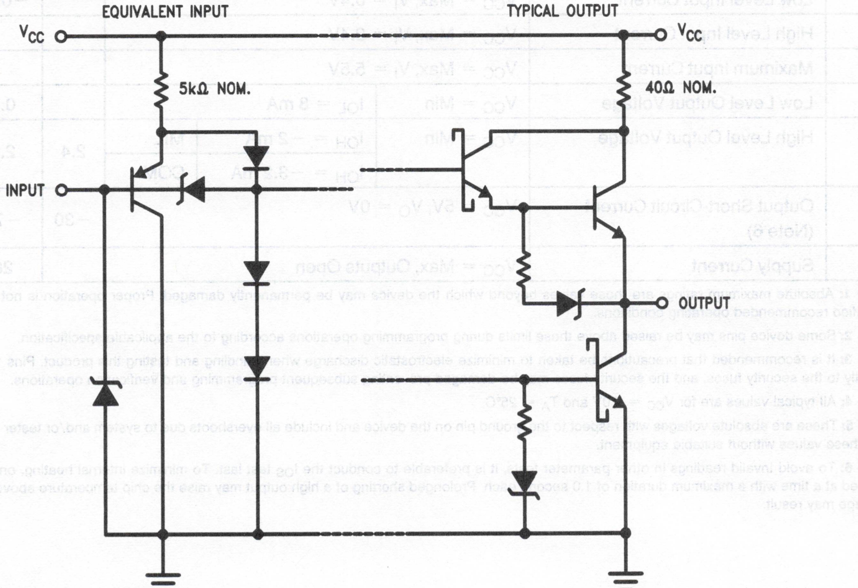
Notes:

V_T = 1.5V

C_L includes probe and jig capacitance.

In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Schematic of Inputs and Outputs


EQUIVALENT INPUT
TYPICAL OUTPUT
V_{CC}
V_{CC}
5kΩ NOM.
40Ω NOM.
INPUT
OUTPUT

TL/L/9995-4

Series A2 (PAL10H8A2, PAL12H6A2, PAL14H4A2, PAL16H2A2, PAL10L8A2, PAL12L6A2, PAL14L4A2, PAL16L2A2, PAL16C1A2)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5 to +7.0V
Input Voltage (Notes 2 and 3)	−1.5 to +5.5V
Off-State Output Voltage (Note 2)	−1.5 to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	Military			Commercial			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature	−55			0		75	°C
T_C	Operating Case Temperature			125				°C

Electrical Characteristics Over Recommended Operating Conditions (Note 4)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 5)						0.8	V
V_{IH}	High Level Input Voltage (Note 5)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 8 mA			0.3	0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OS}	Output Short-Circuit Current (Note 6)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				28	45	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: All typical values are for V_{CC} = 5.0V and T_A = 25°C.

Note 5: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

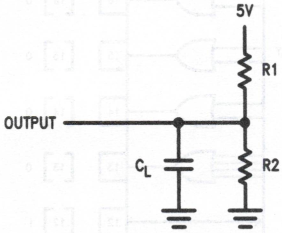
Note 6: To avoid invalid readings in other parameter tests, it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 second each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Series A2 (PAL10H8A2, PAL12H6A2, PAL14H4A2, PAL16H2A2, PAL10L8A2, PAL12L6A2, PAL14L4A2, PAL16L2A2, PAL16C1A2) (Continued)

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions		Military			Commercial			Units
				Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input to Output	C_L = 50 pF	Except 16C1A2		25	45		25	35	ns
			16C1A2			45			40	ns

Test Load


5V
R1
OUTPUT
C_L
R2

MIL/COM

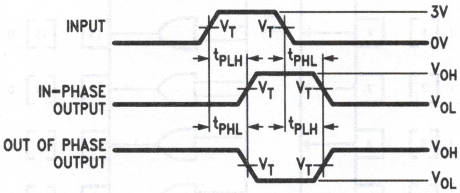
R1 = 560

R2 = 1.1K

TL/L/9995-2

Test Waveform

Propagation Delay


3V
0V
INPUT
V_T
V_T
t_{PLH}
t_{PHL}
V_{OH}
V_{OL}
IN-PHASE OUTPUT
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT
V_{OH}
V_{OL}

TL/L/9995-3

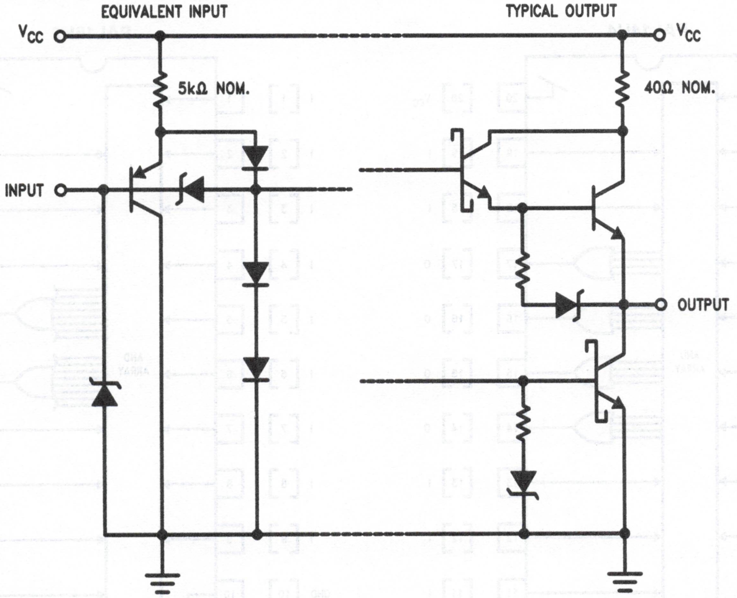
Notes:

V_T = 1.5V

C_L includes probe and jig capacitance.

In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

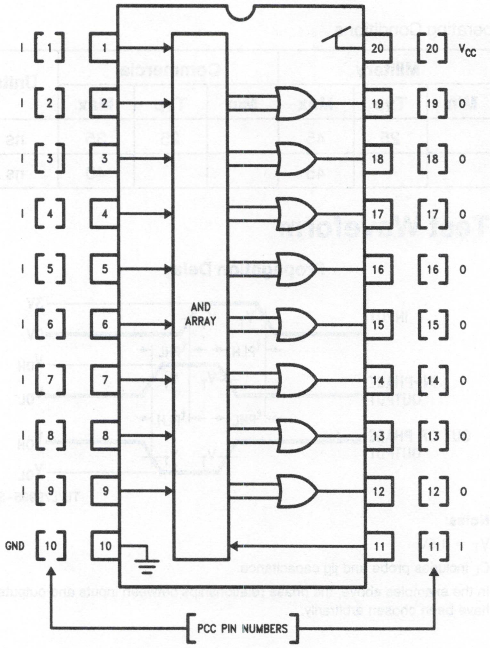
Schematic of Inputs and Outputs


EQUIVALENT INPUT
TYPICAL OUTPUT
V_{CC}
V_{CC}
5kΩ NOM.
40Ω NOM.
INPUT
OUTPUT

TL/L/9995-4

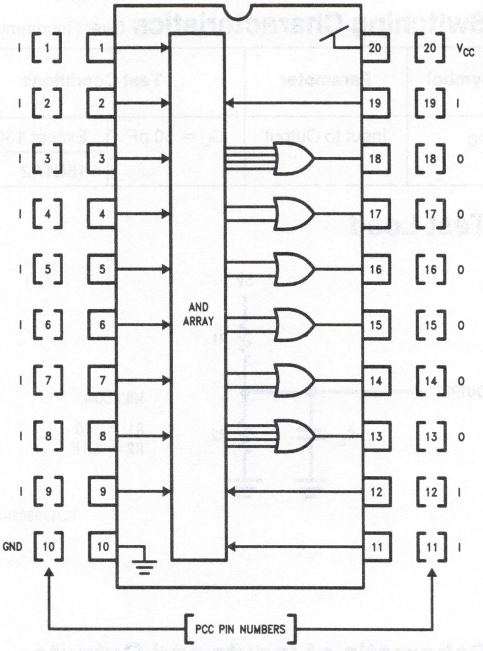
20-Pin Small PAL Family Block Diagrams—DIP Connections

PAL10H8


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 O
18 18 O
17 17 O
16 16 O
15 15 O
14 14 O
13 13 O
12 12 O
11 11 I
PCC PIN NUMBERS

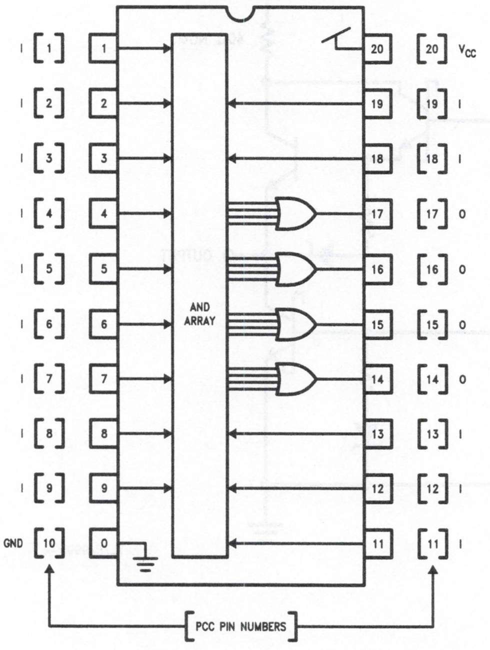
TL/L/9995-5

PAL12H6


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 I
18 18 O
17 17 O
16 16 O
15 15 O
14 14 O
13 13 O
12 12 I
11 11 I
PCC PIN NUMBERS

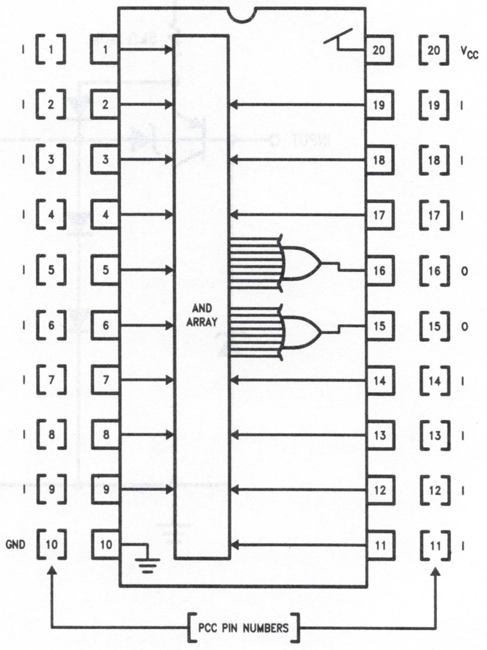
TL/L/9995-6

PAL14H4


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 0
AND ARRAY
20 20 VCC
19 19 I
18 18 I
17 17 O
16 16 O
15 15 O
14 14 O
13 13 I
12 12 I
11 11 I
PCC PIN NUMBERS

TL/L/9995-7

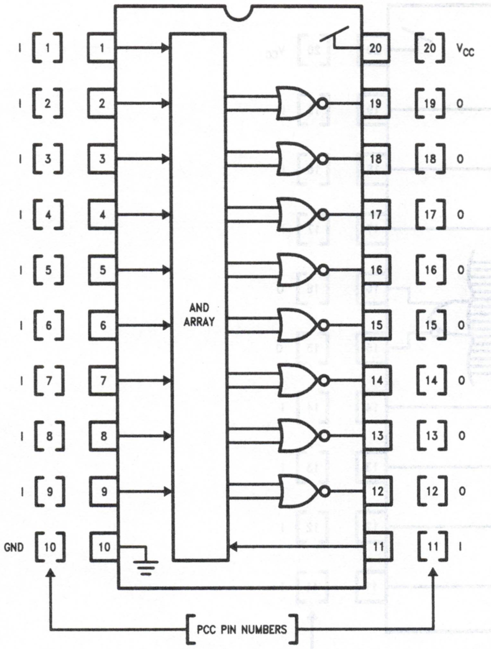
PAL16H2


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 I
18 18 I
17 17 I
16 16 O
15 15 O
14 14 I
13 13 I
12 12 I
11 11 I
PCC PIN NUMBERS

TL/L/9995-8

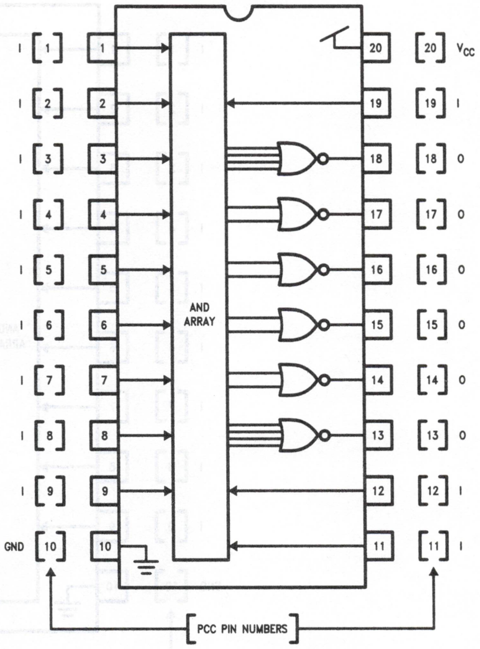
20-Pin Small PAL Family Block Diagrams—DIP Connections (Continued)

PAL10L8


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 O
18 18 O
17 17 O
16 16 O
15 15 O
14 14 O
13 13 O
12 12 O
11 11 I
PCC PIN NUMBERS

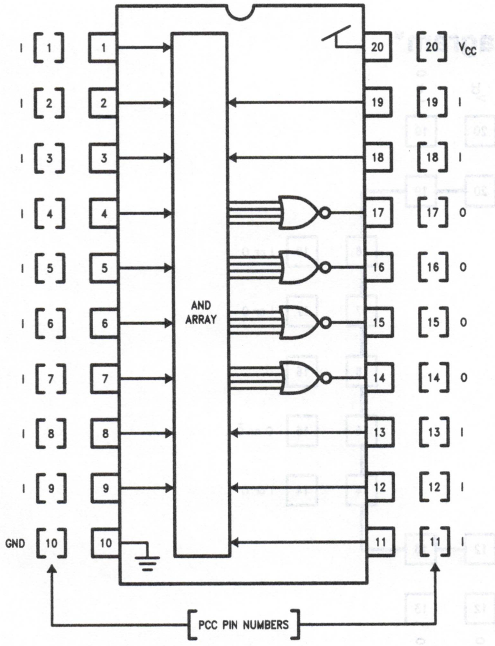
TL/L/9995-9

PAL12L6


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 I
18 18 O
17 17 O
16 16 O
15 15 O
14 14 O
13 13 O
12 12 I
11 11 I
PCC PIN NUMBERS

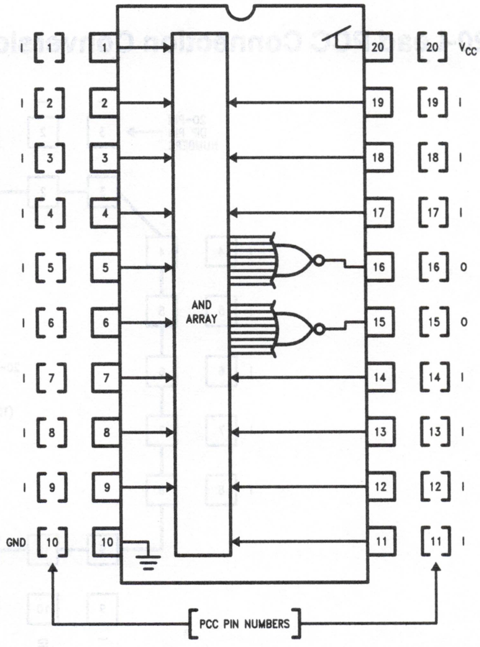
TL/L/9995-10

PAL14L4


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 I
18 18 I
17 17 O
16 16 O
15 15 O
14 14 O
13 13 I
12 12 I
11 11 I
PCC PIN NUMBERS

TL/L/9995-11

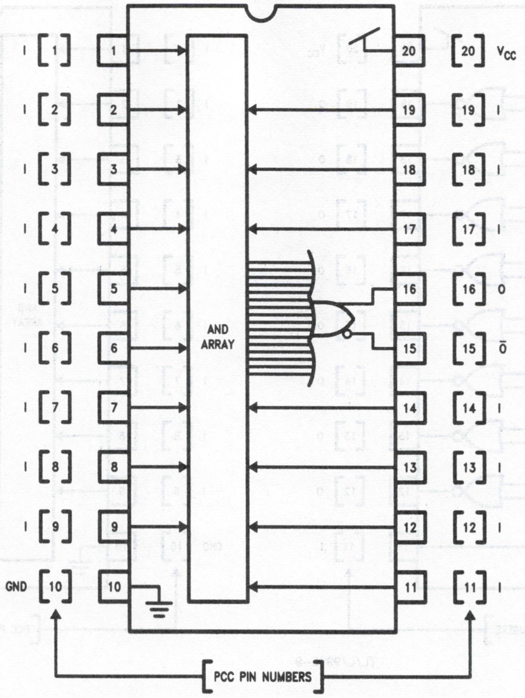
PAL16L2


I 1 1
I 2 2
I 3 3
I 4 4
I 5 5
I 6 6
I 7 7
I 8 8
I 9 9
GND 10 10
AND ARRAY
20 20 VCC
19 19 I
18 18 I
17 17 I
16 16 O
15 15 O
14 14 I
13 13 I
12 12 I
11 11 I
PCC PIN NUMBERS

TL/L/9995-12

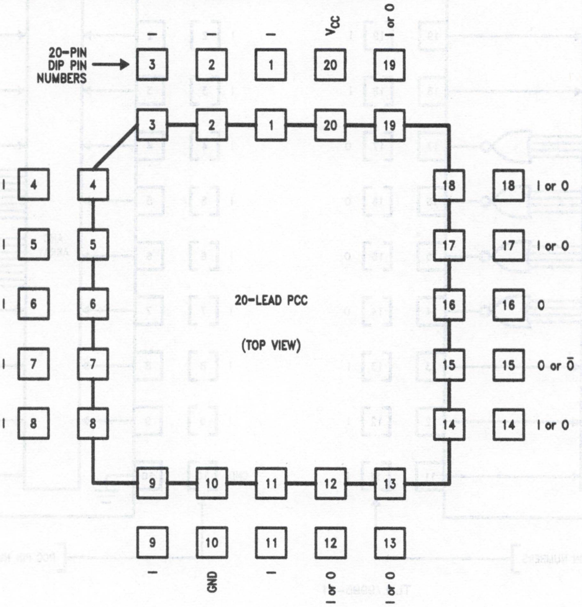
20-Pin Small PAL Family Block Diagrams—DIP Connections (Continued)

PAL16C1


I 1
I 2
I 3
I 4
I 5
I 6
I 7
I 8
I 9
GND 10
20 V_{CC}
19 I
18 I
17 I
16 O
15 $\overline{O}$
14 I
13 I
12 I
11 I
AND ARRAY
PCC PIN NUMBERS

TL/L/9995-13

20-Lead PCC Connection Conversion Diagram*


20-PIN DIP PIN NUMBERS
I 3
I 2
I 1
V_{CC} 20
I or O 19
I 4
I 5
I 6
I 7
I 8
18 I or O
17 I or O
16 O
15 O or $\overline{O}$
14 I or O
20-LEAD PCC
(TOP VIEW)
9 I
10 GND
11 I
12 I or O
13 I or O

TL/L/9995-14

*Series-A parts are not available in this package.

Functional Description

The 20-pin Small PAL logic arrays consist of between 10 and 16 complementary input lines and 16 product-term lines with a programmable fuse link at each intersection (up to 512 fuses). The family consists of nine device types with different numbers of combinatorial outputs. The 20-pin Small PAL Family Block Diagrams show the number of product terms allocated to each output for each device. All product terms allocated to each output connect into an OR-gate to produce the sum-of-products output logic function.

An unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses) are in the high logic state. Therefore, if both the true and complement of at least one array input is left connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed device). Conversely, if all fuses on a product term were programmed, the product term and the resulting logic function would be held in the high state.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

Security Fuse

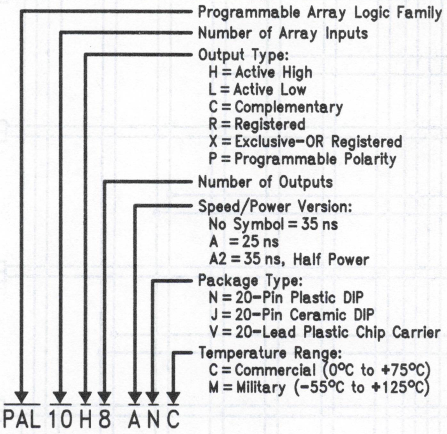
Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses.

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a variety of programming equipment. Many software packages and programming units support a wide variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

Detailed logic diagrams showing all JEDEC fuse-map addresses for the 20-pin Small PAL family are provided for direct map editing and diagnostic purposes. Contact your local National Semiconductor sales representative or distributor for a list of current software and programming support tools available for these devices. Contact the National Semiconductor Programmable Device Support Department if detailed specifications of PAL programming algorithms are needed.

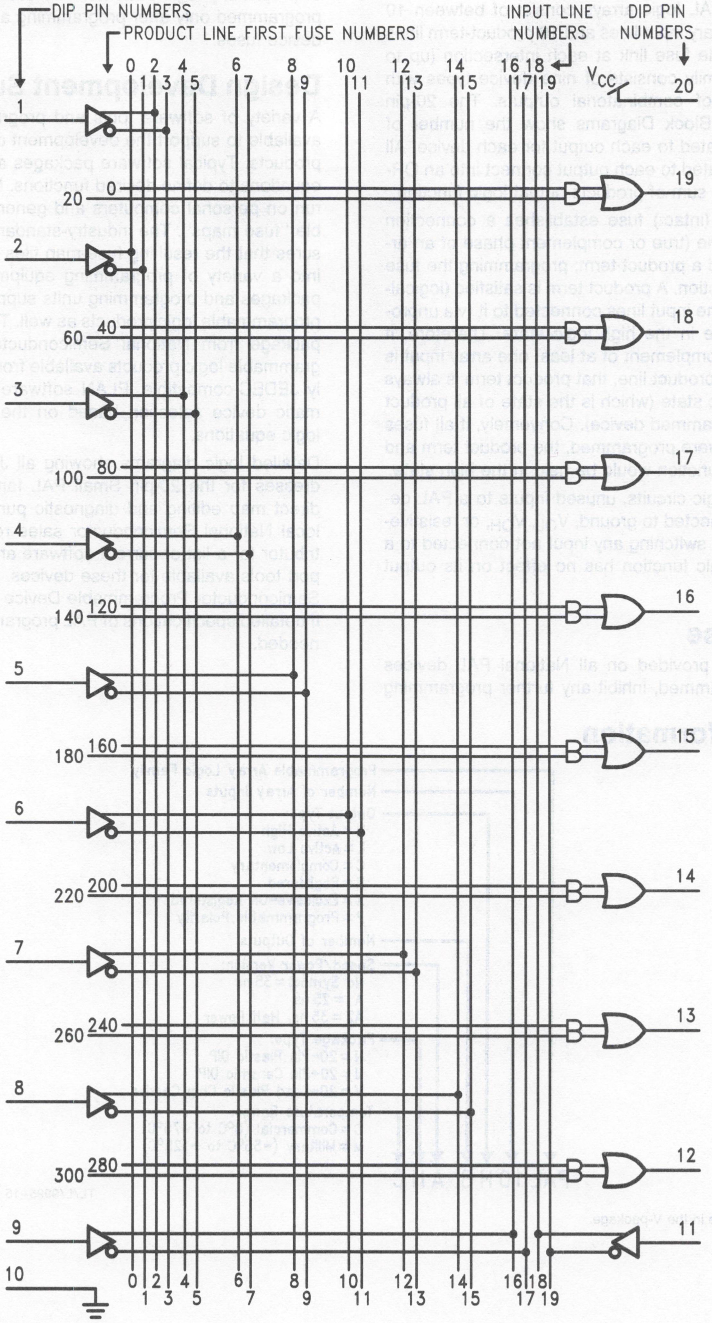
Ordering Information


Programmable Array Logic Family
Number of Array Inputs
Output Type:
H = Active High
L = Active Low
C = Complementary
R = Registered
X = Exclusive-OR Registered
P = Programmable Polarity
Number of Outputs
Speed/Power Version:
No Symbol = 35 ns
A = 25 ns
A2 = 35 ns, Half Power
Package Type:
N = 20-Pin Plastic DIP
J = 20-Pin Ceramic DIP
V = 20-Lead Plastic Chip Carrier
Temperature Range:
C = Commercial (0°C to +75°C)
M = Military (−55°C to +125°C)
PAL 10 H 8 A N C

TL/L/9995-15

*Series-A is not available in the V-package.

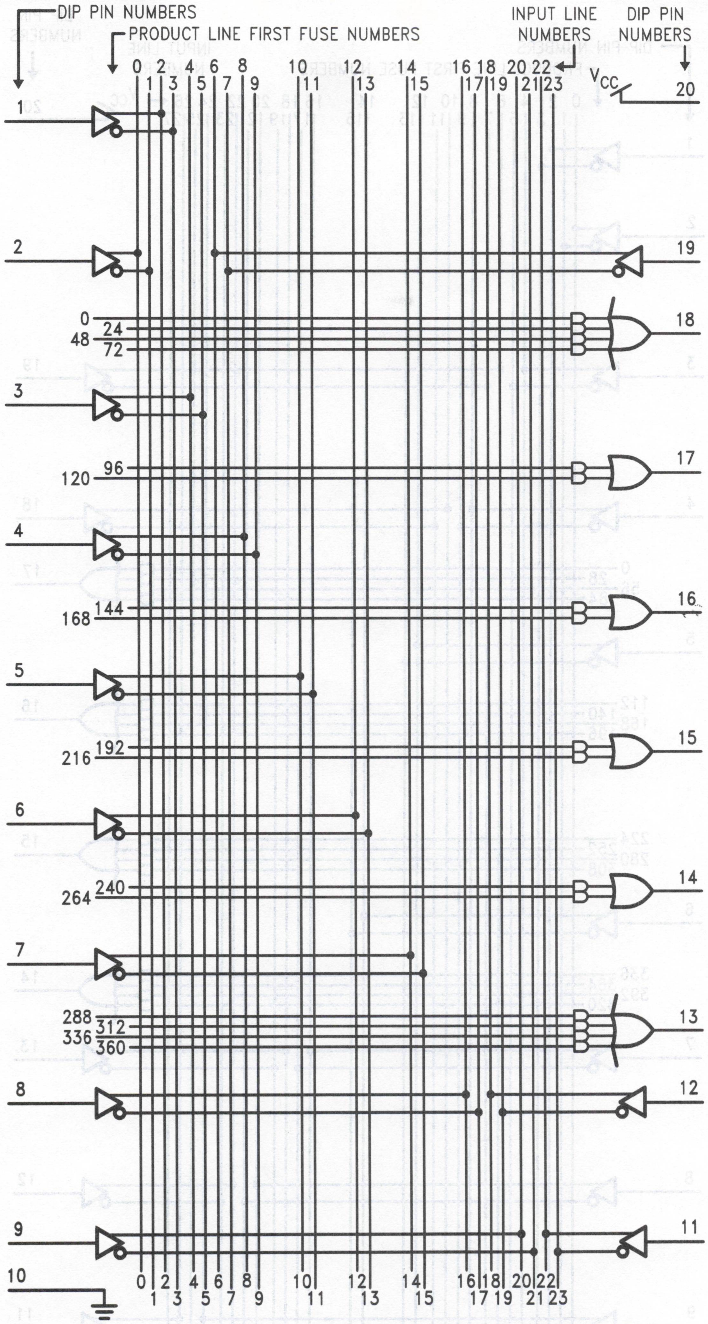
Logic Diagram PAL10H8


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 2 4 6 8 10 12 14 16 18
1 3 5 7 9 11 13 15 17 19
V_{CC}
20
1
20
0
19
2
60
40
18
3
100
80
17
4
140
120
16
5
180
160
15
6
220
200
14
7
260
240
13
8
300
280
12
9
11
10
0 2 4 6 8 10 12 14 16 18
1 3 5 7 9 11 13 15 17 19

TL/L/9995-16

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

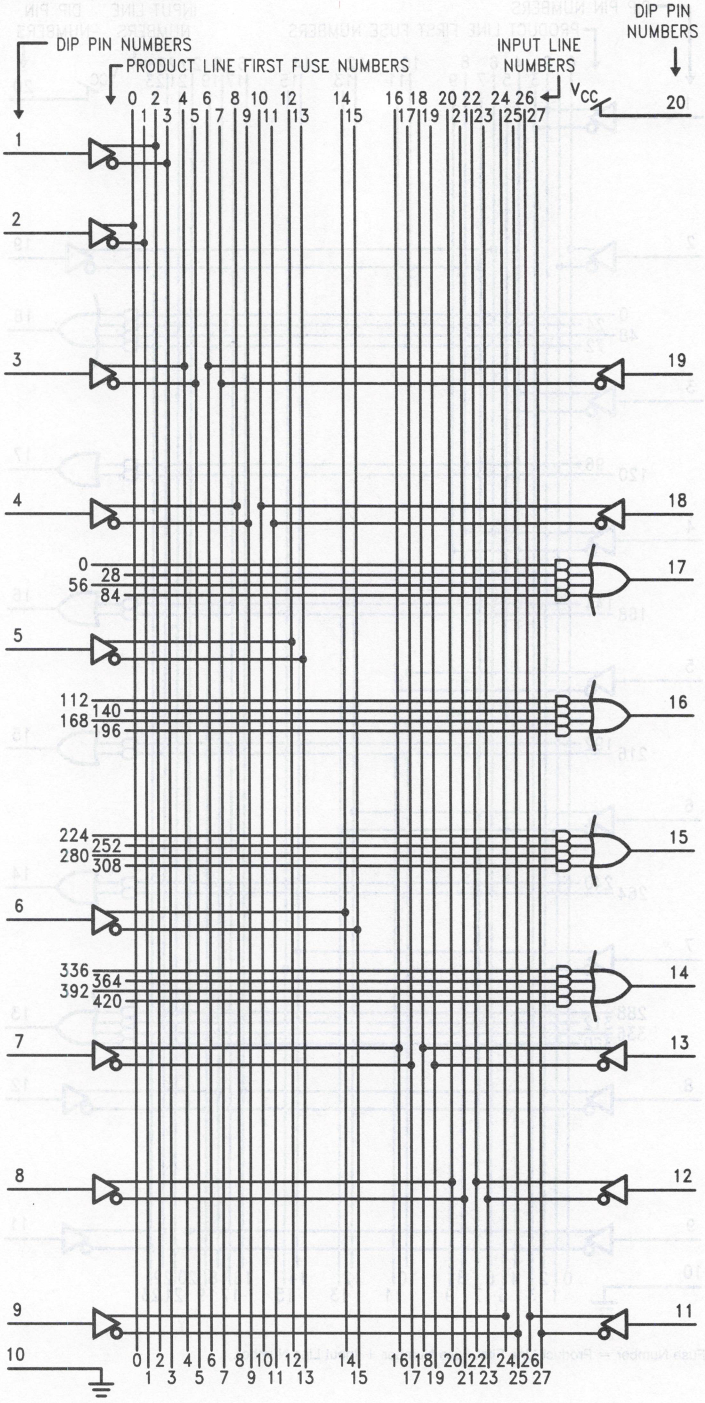
Logic Diagram PAL12H6


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 2 4 6 8 10 12 14 16 18 20 22
1 3 5 7 9 11 13 15 17 19 21 23
V_{CC}
20
1
2
19
0
24
48
72
18
3
96
120
17
4
144
168
16
5
192
216
15
6
240
264
14
7
288
312
336
360
13
8
12
9
11
10
0 2 4 6 8 10 12 14 16 18 20 22
1 3 5 7 9 11 13 15 17 19 21 23

TL/L/9995-17

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

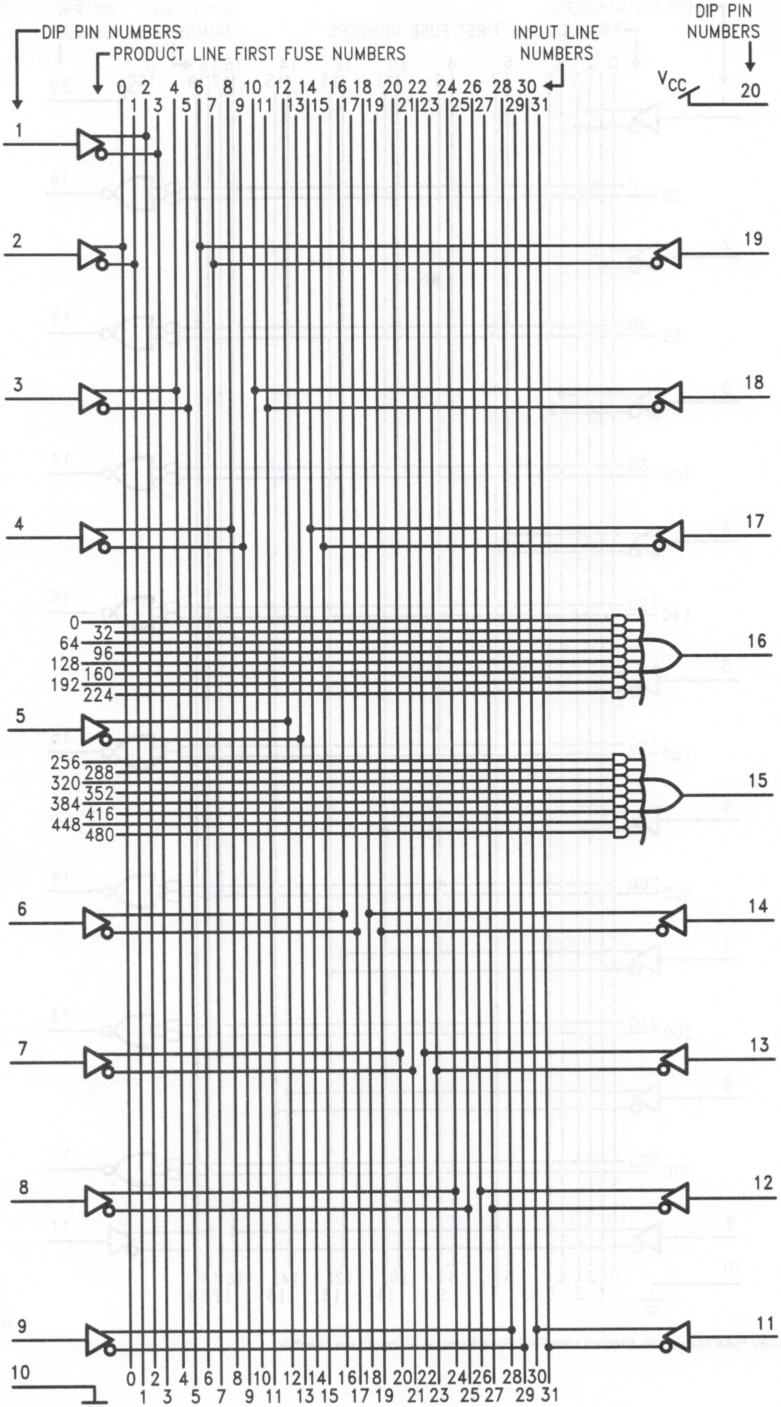
Logic Diagram PAL14H4


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
1
2
3
19
4
18
0
28
56
84
17
5
112
140
168
196
16
224
252
280
308
15
6
336
364
392
420
14
7
13
8
12
9
11
10

TL/L/9995-18

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

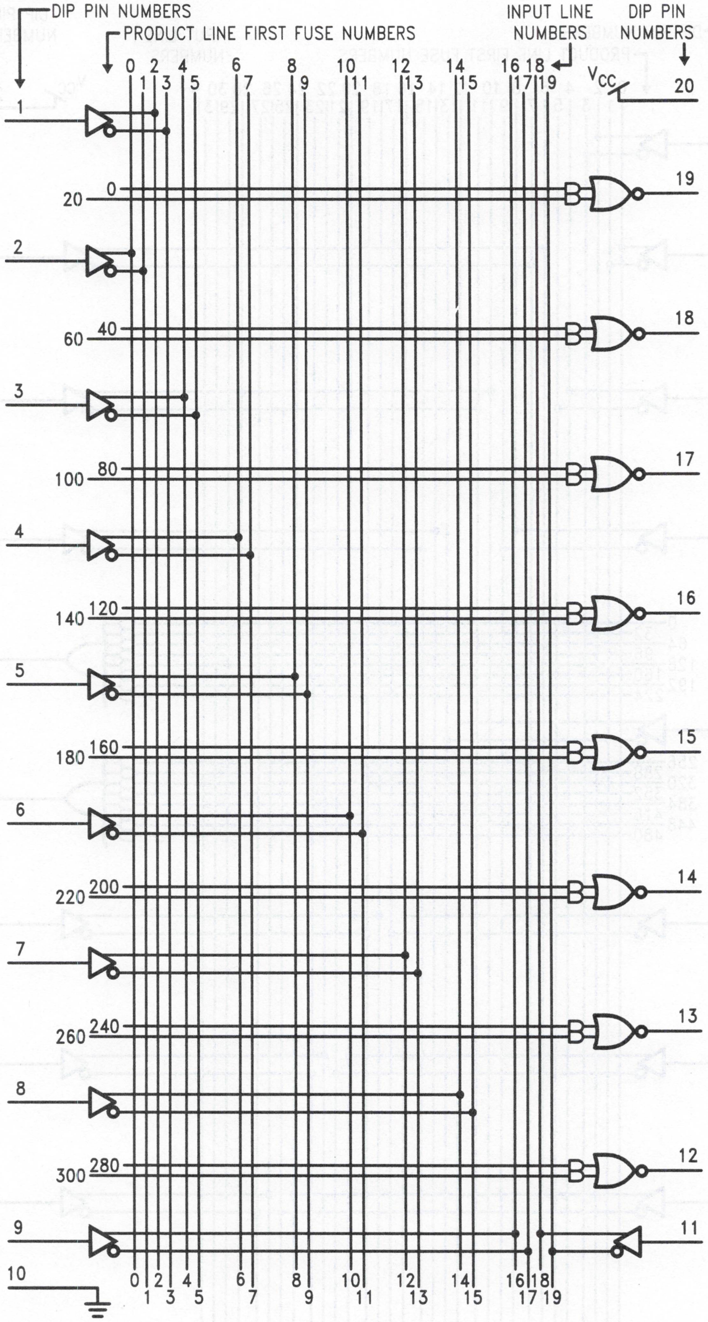
Logic Diagram PAL16H2


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
V_{CC}
20
1
2
19
3
18
4
17
0
32
64
96
128
160
192
224
16
5
256
288
320
352
384
416
448
480
15
6
14
7
13
8
12
9
11
10
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

TL/L/9995-19

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

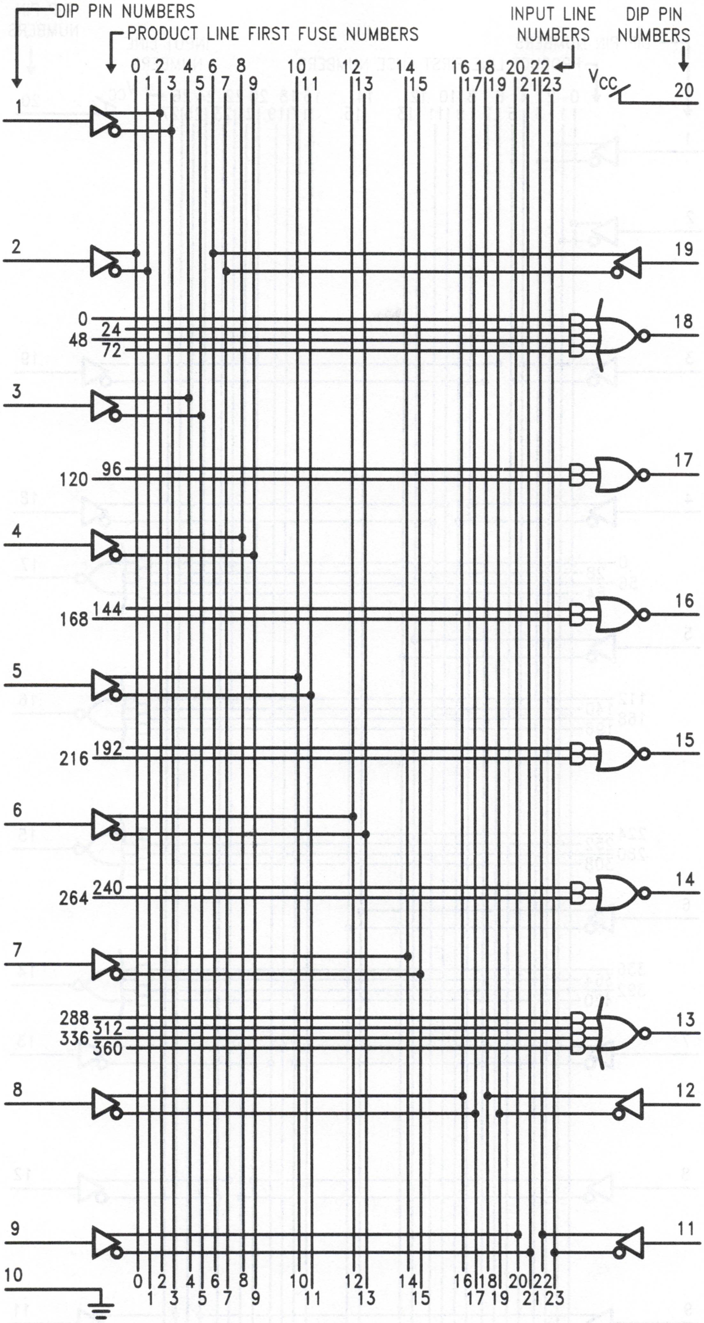
Logic Diagram PAL10L8


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
V_{CC}
20
1
20
0
19
2
60
40
18
3
100
80
17
4
140
120
16
5
180
160
15
6
220
200
14
7
260
240
13
8
300
280
12
9
11
10

TL/L/9995-20

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

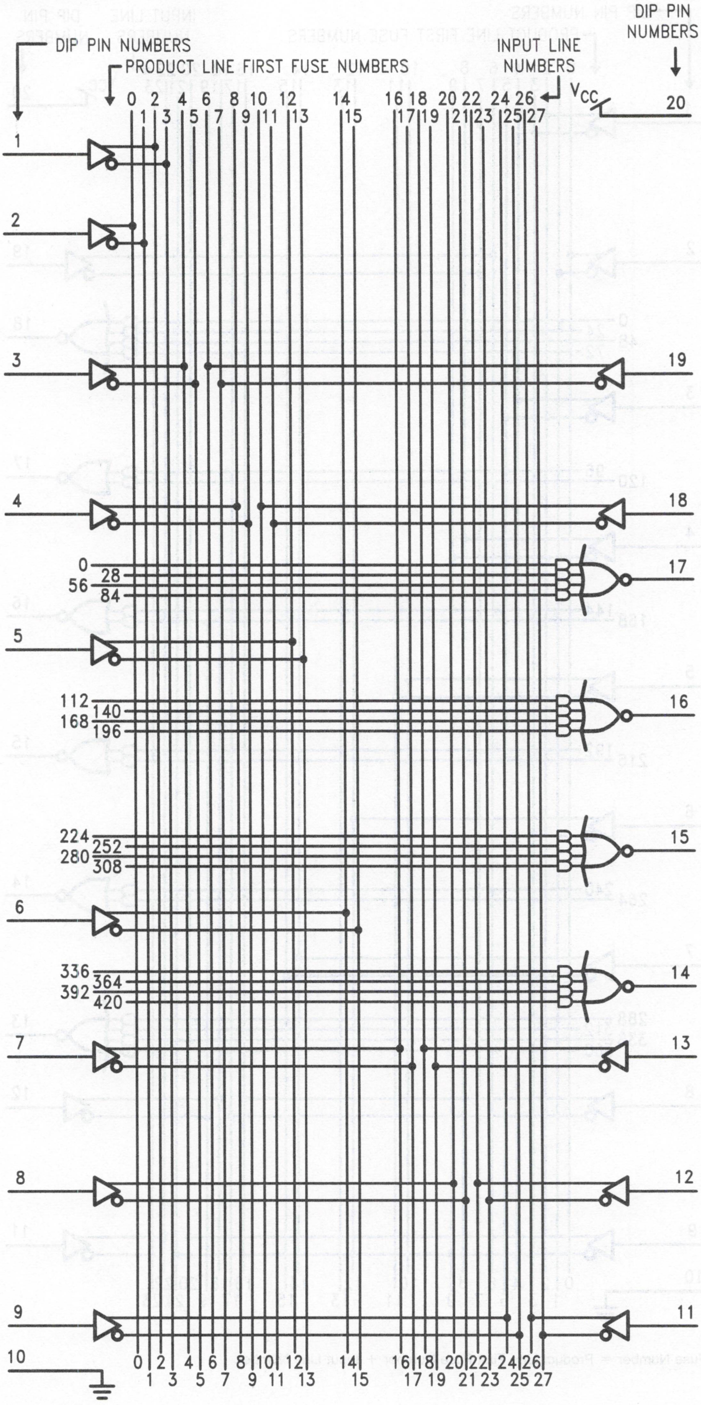
Logic Diagram PAL12L6


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
VCC
20
1
2
19
0
24
48
72
18
3
120
96
17
4
168
144
16
5
216
192
15
6
264
240
14
7
288
312
336
360
13
8
12
9
11
10

TL/L/9995-21

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

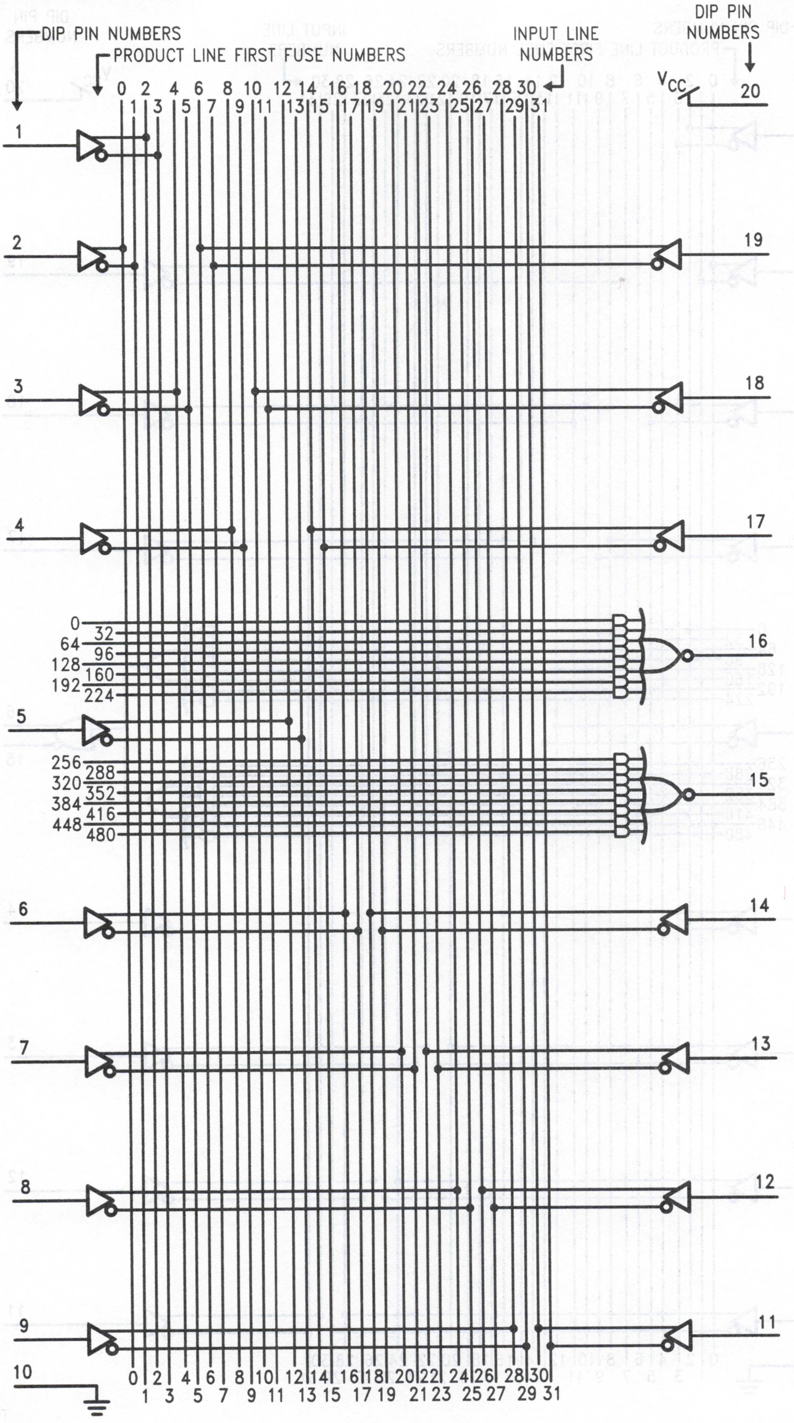
Logic Diagram PAL14L4


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
1
2
3
19
4
18
0
28
56
84
17
5
112
140
168
196
16
224
252
280
308
15
6
336
364
392
420
14
7
13
8
12
9
11
10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

TL/L/9995-22

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

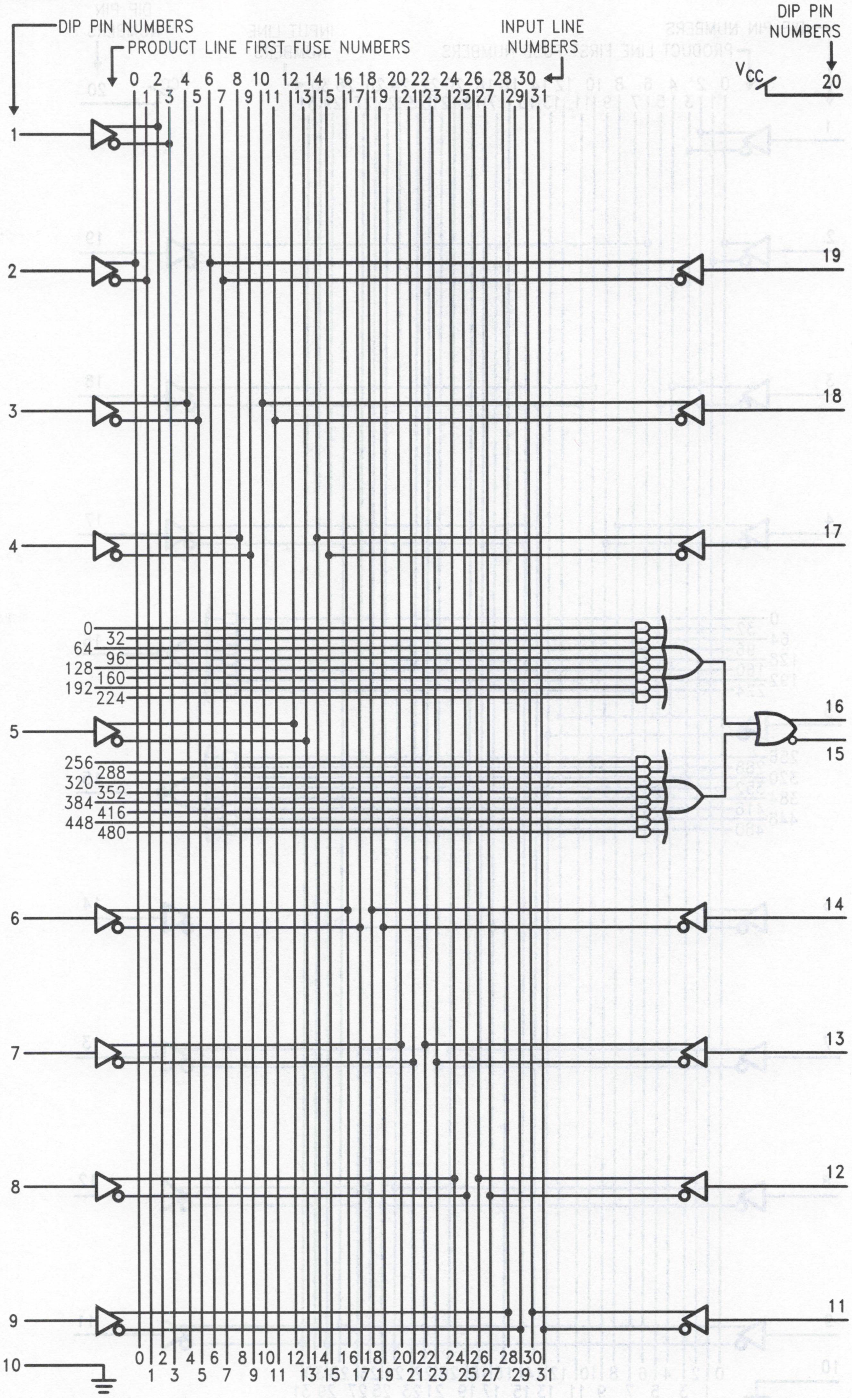
Logic Diagram PAL16L2


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
20
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
1
2
19
3
18
4
17
0
32
64
96
128
160
192
224
16
5
256
288
320
352
384
416
448
480
15
6
14
7
13
8
12
9
11
10

TL/L/9995–23

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

Logic Diagram PAL16C1


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
20
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
1
2
3
4
5
6
7
8
9
10
19
18
17
16
15
14
13
12
11
0
32
64
96
128
160
192
224
256
288
320
352
384
416
448
480

TL/L/9995-24

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

National Semiconductor

Programmable Array Logic (PAL®) 20-Pin Medium PAL Family

General Description

The 20-pin Medium PAL family contains four of the most popular PAL architectures used in industry. National Semiconductor's advanced Schottky TTL process with titanium tungsten fusible links is used in manufacturing the standard, Series-A, Series-A2, Series-B and Series-B2 devices. Series-D devices are manufactured using National Semiconductor's isoplanar "FAST-Z" TTL process with highly reliable "vertical-fuse" programmable cells. Vertical fuses are implemented using avalanche-induced migration ("AIM") technology offering very high programming yields and is an extension of National's FAST logic family. The 20-pin Medium PAL Family provides high-speed user-programmable replacements for conventional SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming the programmable cells to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

The PAL logic array has a total of 16 complementary input pairs and 8 outputs generated by a single programmable AND-gate array with fixed OR-gate connections. Device outputs are either taken directly from the AND-OR functions (combinatorial) or passed through D-type flip-flops (registered). Registers allow the PAL device to implement sequential logic circuits. TRI-STATE® outputs facilitate busing and provide bidirectional I/O capability. The medium PAL family offers a variety of combinatorial and registered output mixtures, as shown in the Device Types table below.

On power-up, Series-D devices reset all registers to simplify sequential circuit design and testing. For these devices, direct register preload is also provided to facilitate device testing. Security fuses can be programmed to prevent direct copying of proprietery logic patterns in all the family devices.

Features

- As fast as 10 ns maximum propagation delay (combinatorial)
- User-programmable replacement for TTL logic
- High programming yield and reliability of vertical-fuse technology for Series-D products. (Programming equipment with certified vertical-fuse algorithm required.)
- Extension of FAST product line (Series-D).
- Large variety of JEDEC-compatible programming equipment and design development software available
- Fully supported by National PLAN™ development software
- Power-up reset for registered outputs (Series-D)
- Register preload facilitates device testing (Series-B,D)
- Security fuse prevents direct copying of logic patterns

Device Types

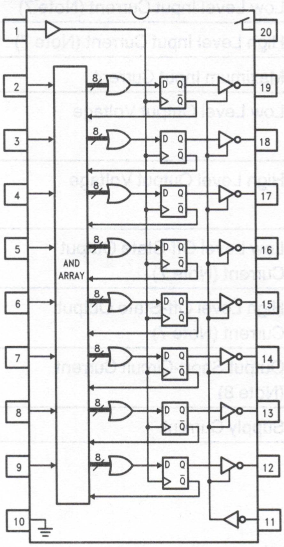
Device Type	Dedicated Inputs	Registered Outputs (with Feedback)	Combinatorial	
			I/Os	Outputs
PAL16L8	10	—	6	2
PAL16R4	8	4	4	—
PAL16R6	8	6	2	—
PAL16R8	8	8		

Speed/Power Versions

Series	Example	Commercial		Military	
		t_{PD}	I_{CC}	t_{PD}	I_{CC}
Standard	PAL16L8	35 ns	180 mA	45 ns	180 mA
A	PAL16L8A	25 ns	180 mA	30 ns	180 mA
A2	PAL16L8A2	35 ns	90 mA	50 ns	90 mA
B	PAL16L8B	15 ns	180 mA	20 ns	180 mA
B2	PAL16L8B2	25 ns	90 mA*	30 ns	90 mA*
D	PAL16L8D	10 ns	180 mA		

*For the registered devices I_{CC} = 100 mA.

Block Diagram—PAL16R8



TL/L/9391-1

Standard Series (PAL16L8, PAL16R4, PAL16R6, PAL16R8)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Notes 2 and 3)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter		Military Min	Military Nom	Military Max	Commercial Min	Commercial Nom	Commercial Max	Units
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	25			25			ns
		High	25			25			ns
t_{SU}	Setup Time from Input or Feedback to Clock		45			35			ns
t_H	Hold Time of Input after Clock		0	−15		0	−15		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		25	12.2		25	16.6	MHz
		Without Feedback		35	20		35	20	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				120	180	mA

Standard Series (PAL16L8, PAL16R4, PAL16R6, PAL16R8) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged, preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

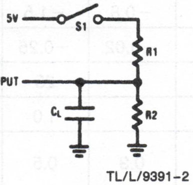
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a High output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

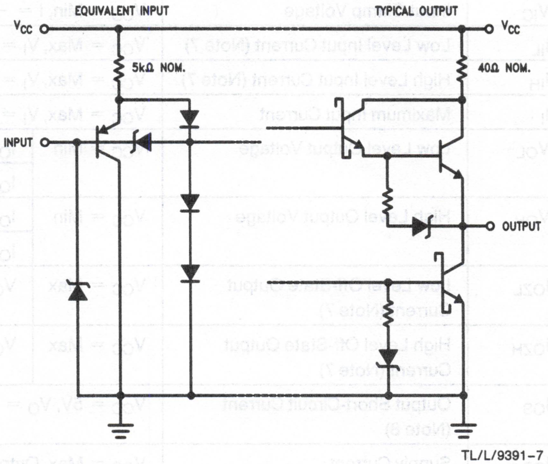
Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		25	45		25	35	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		15	25		15	25	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		15	25		15	25	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		15	25		15	25	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		25	45		25	35	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		25	45		25	35	ns

Test Load



MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

Schematic of Inputs and Outputs



Series-A (PAL16L8A, PAL16R4A, PAL16R6A, PAL16R8A)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Notes 2 and 3)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	20	10		15	10		ns
		High	20	10		15	10		ns
t_{SU}	Setup Time from Input or Feedback to Clock		30	16		25	16		ns
t_H	Hold Time of Input after Clock		0	−10		0	−10		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		38.5	20		38.5	25	MHz
		Without Feedback		50	25		50	33.3	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				120	180	mA

Series-A (PAL16L8A, PAL16R4A, PAL16R6A, PAL16R8A) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged, preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

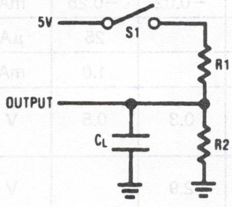
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a High output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		15	30		15	25	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		10	20		10	15	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	25		10	20	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		11	25		11	20	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	30		10	25	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		13	30		13	25	ns

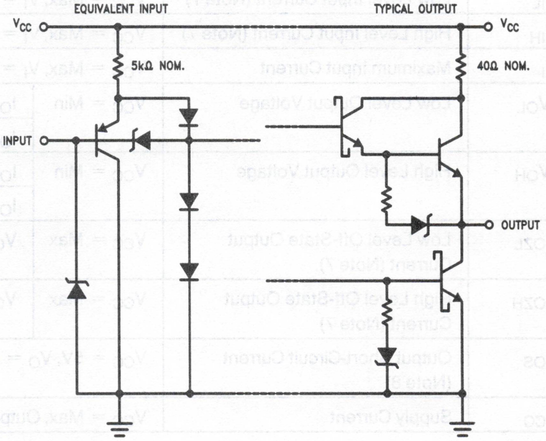
Test Load



MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

TL/L/9391-2

Schematic of Inputs and Outputs



TL/L/9391-7

Series-A2 (PAL16L8A2, PAL16R4A2, PAL16R6A2, PAL16R8A2)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Notes 2 and 3)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55		125	0		75	°C
t_W	Clock Pulse Width	Low	25	10		25	10		ns
		High	25	10		25	10		ns
t_{SU}	Setup Time from Input or Feedback to Clock		50	25		35	25		ns
t_H	Hold Time of Input after Clock		0	−15		0	−15		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		25	13.3		25	16.7	MHz
		Without Feedback		50	20		50	20	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				70	90	mA

Series-A2 (PAL16L8A2, PAL16R4A2, PAL16R6A2, PAL16R8A2) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged, preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

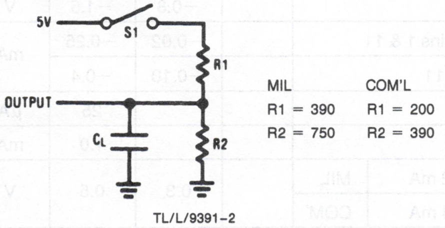
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a High output may raise the chip temperature above normal and permanent damage may result.

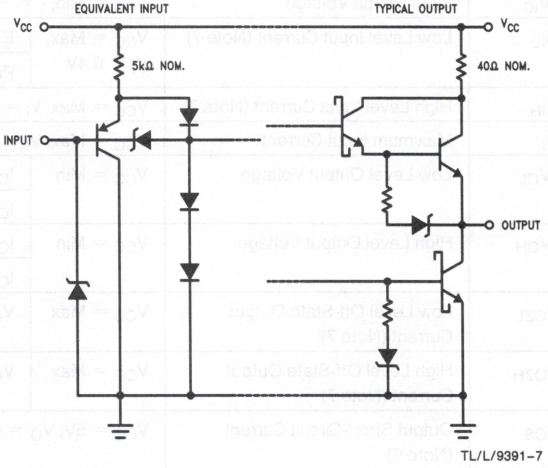
Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		25	50		25	35	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		15	25		15	25	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		15	25		15	25	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		15	25		15	25	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		25	45		25	35	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		25	45		25	35	ns

Test Load



Schematic of Inputs and Outputs



Series-B (PAL16L8B, PAL16R4B, PAL16R6B, PAL16R8B)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Notes 2 and 3)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	12	5		10	5		ns
		High	12	5		10	5		ns
t_{SU}	Setup Time from Input or Feedback to Clock		20	10		15	10		ns
t_H	Hold Time of Input after Clock		0	−5		0	−5		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		55.5	28.6		55.5	40	MHz
		Without Feedback		100	41.7		100	50	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V	Except pins 1 & 11			−0.02	−0.25	mA
			Pins 1 & 11			−0.10	−0.4	
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	3.4		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	µA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	µA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				120	180	mA

Series-B (PAL16L8B, PAL16R4B, PAL16R6B, PAL16R8B) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and the security fuses may be damaged, preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(t_{WLOW} + t_{WHIGH})^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

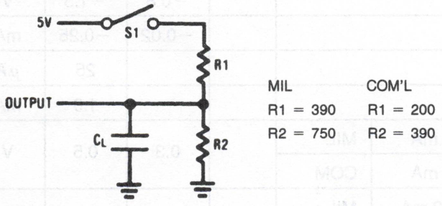
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a High output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

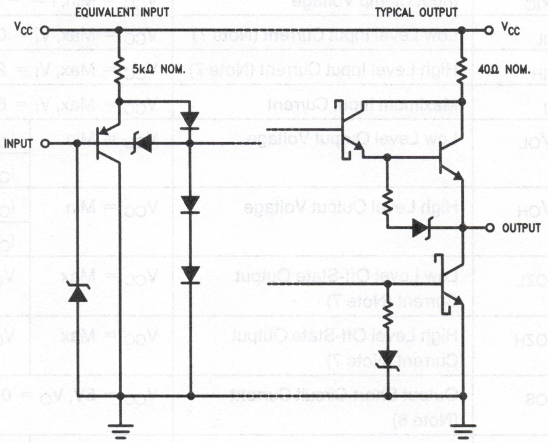
Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		11	20		11	15	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		8	15		8	12	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	20		10	15	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		10	20		10	15	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		11	25		11	20	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		11	20		11	15	ns

Test Load



TL/L/9391-2

Schematic of Inputs and Outputs



TL/L/9391-7

Series-B2 (PAL16L8B2, PAL16R4B2, PAL16R6B2, PAL16R8B2)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance (Note 3)	2000V

$C_{ZAP} = 100$ pF
$R_{ZAP} = 1500\Omega$
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55		125	0		75	°C
t_W	Clock Pulse Width	Low	15	8		10	8		ns
		High	20	10		15	10		ns
t_{SU}	Setup Time from Input or Feedback to Clock		25	11		20	11		ns
t_H	Hold Time of Input after Clock		0	−10		0	−10		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		47.6	22.2		47.6	28.6	MHz
		Without Feedback		55.5	28.6		55.5	40	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	3.4		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open	16L8B2			60	90	mA
			16R4B2, 16R6B2, 16R8B2			70	100	

Series-B2 (PAL16L8B2, PAL16R4B2, PAL16R6B2, PAL16R8B2) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and, although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged, preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(t_{WLOW} + t_{WHIGH})^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

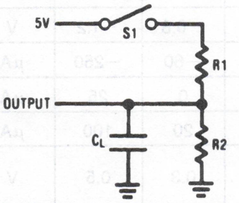
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a High output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	C_L = 50 pF, S1 Closed		15	30		15	25	ns
t_{CLK}	Clock to Registered Output or Feedback	C_L = 50 pF, S1 Closed		10	20		10	15	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		15	25		15	20	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	C_L = 5 pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		11	25		11	20	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		10	30		10	25	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	C_L = 5 pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		13	30		13	25	ns

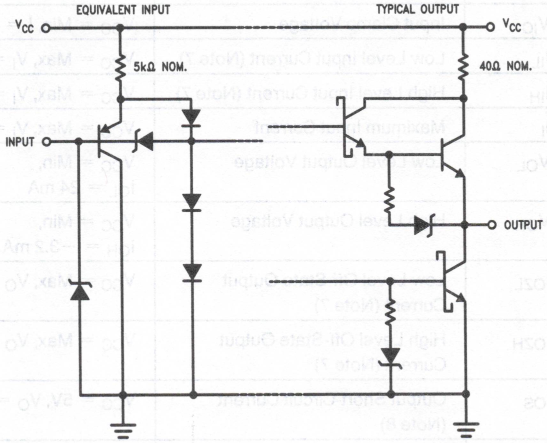
Test Load



MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

TL/L/9391-2

Schematic of Inputs and Outputs



TL/L/9391-7

Series-D (PAL16L8D, PAL16R4D, PAL16R6D, PAL16R8D)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +7.0V
Off-State Output Voltage (V_O) (Notes 2 & 3)	−1.5V to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	+100 mA

Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance	1500V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Commercial			Units
			Min	Nom	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	V
T_A	Operating Free-Air Temperature		0	25	75	°C
t_W	Clock Pulse Width	Low	7	3.5		ns
		High	7	2		ns
t_{SU}	Setup Time from Input or Feedback to Clock		10	5.5		ns
t_H	Hold Time of Input after Clock		0	−3.7		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		90	55.5	MHz
		Without Feedback		120	71	MHz
V_Z	Register Preload Control Voltage		9.5	9.75	10.0	V

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)				0.8	V
V_{IH}	High Level Input Voltage (Note 6)		2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA		−0.8	−1.2	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V		−60	−250	μA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V		0	25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V		20	100	μA
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = 24 mA		0.3	0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = −3.2 mA	2.7	3.1		V
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max, V_O = 0.4V		0	−50	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max, V_O = 2.4V		0	50	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V	−50	−77	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open		125	180	mA
C_I	Input Capacitance	V_{CC} = 5.0V, V_I = 2.0V		8		pF
C_O	Output Capacitance	V_{CC} = 5.0V, V_O = 2.0V		8		pF
$C_{I/O}$	I/O Capacitance	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V		8		pF

Series-D (PAL16L8D, PAL16R4D, PAL16R6D, PAL16R8D) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: V_O must not exceed $V_{CC} + 1V$

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

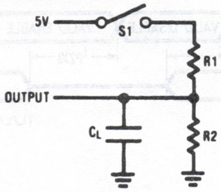
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 second each. Prolonged shorting of a High output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Commercial			Units
			Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	C_L = 50 pF, S1 Closed		7.1	10	ns
t_{CLK}	Clock to Registered Output or Feedback	C_L = 50 pF, S1 Closed		5.5	8	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		5.5	10	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	C_L = 5 pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		4.0	10	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		7.2	10	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	C_L = 5 pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		5.0	10	ns
t_{RESET}	Power-Up to Registered Output High				1000	ns

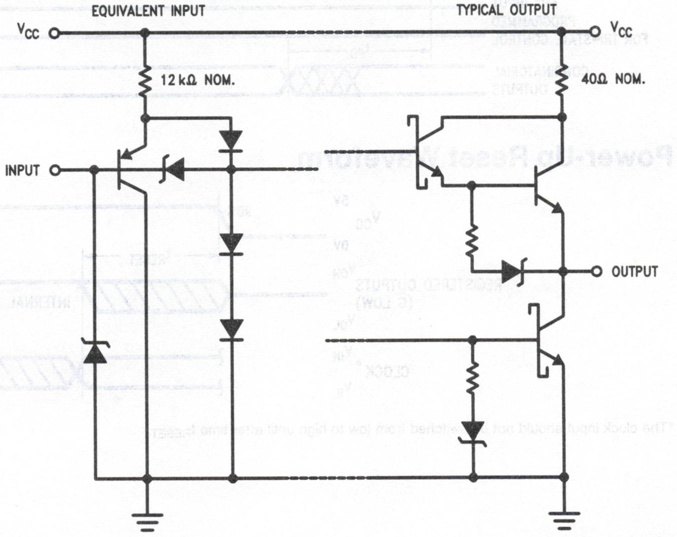
Test Load



TL/L/9391-2

R1 = 200
R2 = 390

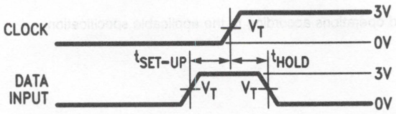
Schematic of Inputs and Outputs



TL/L/9391-24

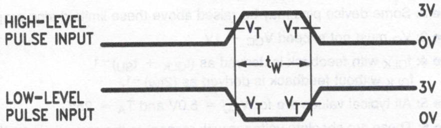
Test Waveforms

Set-Up and Hold


CLOCK
3V
0V
V_T
t_{SET-UP}
t_{HOLD}
DATA INPUT
3V
0V
V_T
V_T

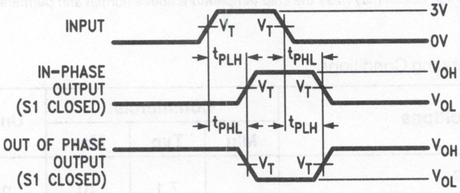
TL/L/9391-3

Pulse Width


HIGH-LEVEL PULSE INPUT
3V
0V
V_T
V_T
t_W
LOW-LEVEL PULSE INPUT
3V
0V
V_T
V_T

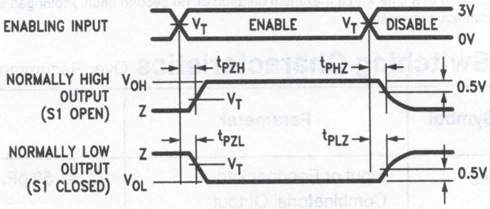
TL/L/9391-4

Propagation Delay


INPUT
3V
0V
V_T
V_T
t_{PLH}
t_{PHL}
IN-PHASE OUTPUT (S1 CLOSED)
V_{OH}
V_{OL}
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT (S1 CLOSED)
V_{OH}
V_{OL}

TL/L/9391-5

Enable and Disable


ENABLING INPUT
V_T
ENABLE
V_T
DISABLE
3V
0V
t_{PZH}
t_{PHZ}
NORMALLY HIGH OUTPUT (S1 OPEN)
V_{OH}
Z
V_T
0.5V
t_{PZL}
t_{PLZ}
NORMALLY LOW OUTPUT (S1 CLOSED)
Z
V_{OL}
V_T
0.5V

TL/L/9391-6

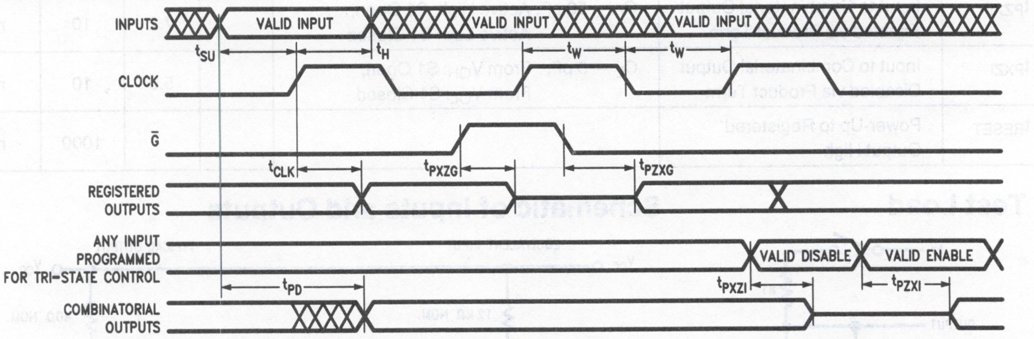
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

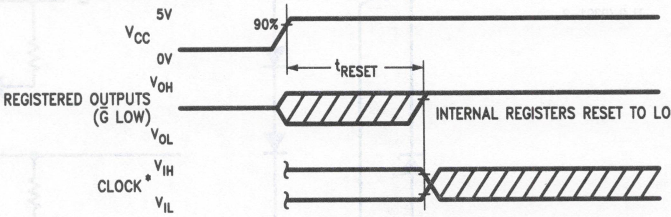
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PXZI}
t_{PZXI}
t_{PD}
COMBINATORIAL OUTPUTS

TL/L/9391-8

Power-Up Reset Waveform


V_{CC}
5V
0V
90%
t_{RESET}
REGISTERED OUTPUTS ($\overline{G}$ LOW)
V_{OH}
V_{OL}
INTERNAL REGISTERS RESET TO LOGIC 0
CLOCK *
V_{IH}
V_{IL}

TL/L/9391-9

*The clock input should not be switched from low to high until after time t_{RESET}.

Functional Description

All of the 20-pin Medium PAL logic arrays consist of 16 complementary input lines and 64 product-term lines with a programmable cell at each intersection (2048 cells). The product terms are organized into eight groups of eight each. Seven or eight of the product terms in each group connect into an OR-gate to produce the sum-of-products logic function, depending on whether the output is combinatorial or registered.

For the fuse-link PAL devices (all PAL devices prior to National Series-D), an unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. In the National Series-D vertical fuse PAL devices, a programmed vertical fuse cell establishes a connection between an input line and a product term. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses for the fuse-link devices, or by programming the corresponding cells for the vertical fuse devices) are in the high logic state. Therefore, if both the true and complement of at least one array input is connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed fuse-link device). Conversely, if all input lines are disconnected from a product line, the product term and the resulting logic function would be held in the high state (which is the state of all product terms in an unprogrammed National Series-D PAL device).

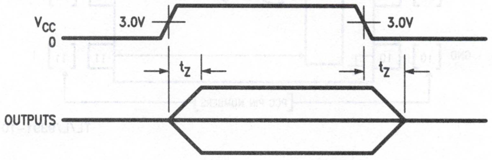
The medium PAL family consists of four device types with differing mixtures of combinatorial and registered outputs. The 16L8, 16R4, 16R6 and 16R8 architectures have 0, 4, 6 and 8 registered outputs, respectively, with the balance of the 8 outputs combinatorial. All outputs are active-low and have TRI-STATE capability.

Each combinatorial output has a seven product-term logic function, with the eighth product term being used for TRI-STATE control. A combinatorial output is enabled while the TRI-STATE product term is satisfied (true). Combinatorial outputs also have feedback paths from the device pins into the logic array (except for two outputs on the 16L8). This allows a pin to perform bidirectional I/O or, if the associated TRI-STATE control product term were programmed to remain unsatisfied (always false), the output driver would remain disabled and the pin could be used as an additional dedicated input.

Registered outputs each have an eight product-term logic function feeding into a D-type flip-flop. All registers are triggered by the high-going edge of the clock input pin. All registered outputs are controlled by a common output enable ($\overline{G}$) pin (enabled while low). The output of each register is also fed back into the logic array via an internal path. This provides for sequential logic circuits (state machines, counters, etc.) which can be sequenced even while the outputs are disabled.

Series-D Medium PAL devices reset all registers to a low state upon power-up (active-low outputs assume high logic levels if enabled). This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

For the National Series-D PAL device, during power-up, all outputs are held in the high-impedance state until DC power supply conditions are met (V_{CC} approximately 3.0V), after which they may be enabled by the TRI-STATE control product terms (combinatorial outputs) or the $\overline{G}$ pin (registered outputs). Whenever V_{CC} goes below 3V (at 25°C), the outputs are disabled as shown in *Figure 1* below.



TL/L/9391-25

Note: t_Z is less than 100 ns.

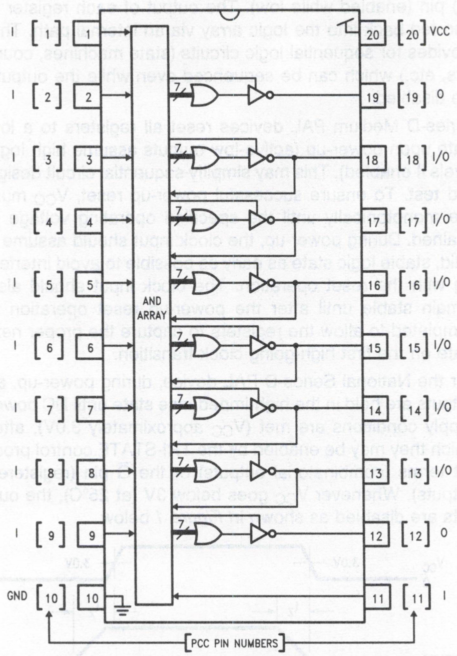
FIGURE 1. Series-D Power-Up TRI-STATE Waveform

In an unprogrammed National Series-D PAL device, no array inputs are connected to any product-term lines. Therefore, all combinatorial outputs would be enabled and driving low logic levels (after power-up is completed). All registers would still initialize to the low state, but would become permanently set (low-level outputs, if enabled) following the first clock transition.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

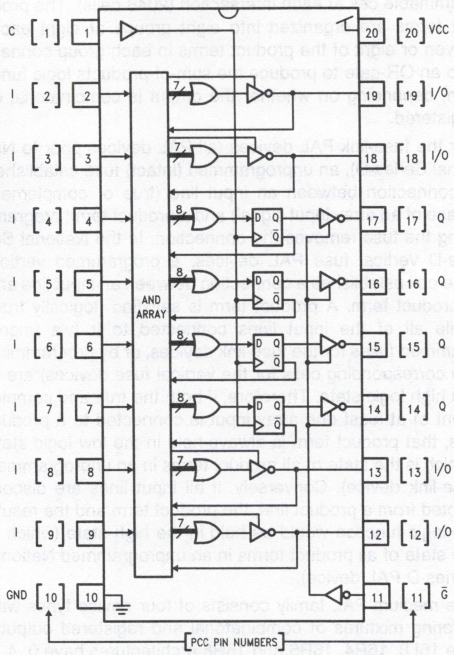
20-Pin Medium PAL Family Block Diagrams—DIP Connections

PAL16L8


I
VCC
O
I/O
GND
AND ARRAY
7
PCC PIN NUMBERS

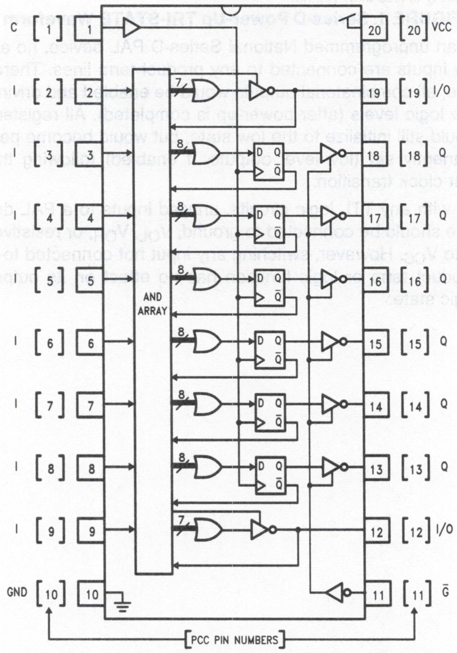
TL/L/9391-10

PAL16R4


C
I
VCC
I/O
Q
GND
G̅
AND ARRAY
7
8
D Q
Q̅
PCC PIN NUMBERS

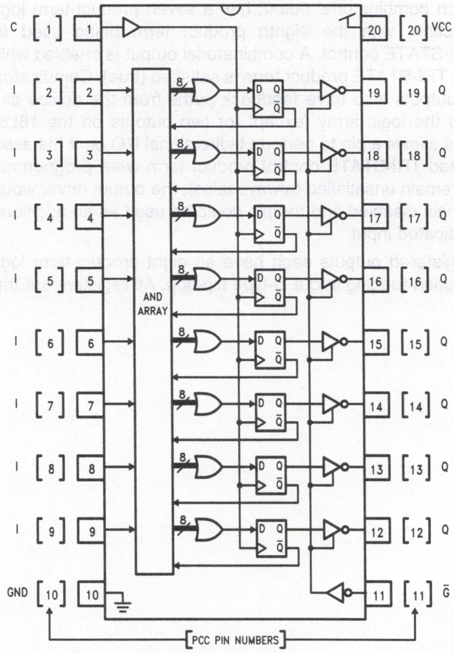
TL/L/9391-11

PAL16R6


C
I
VCC
I/O
Q
GND
G̅
AND ARRAY
7
8
D Q
Q̅
PCC PIN NUMBERS

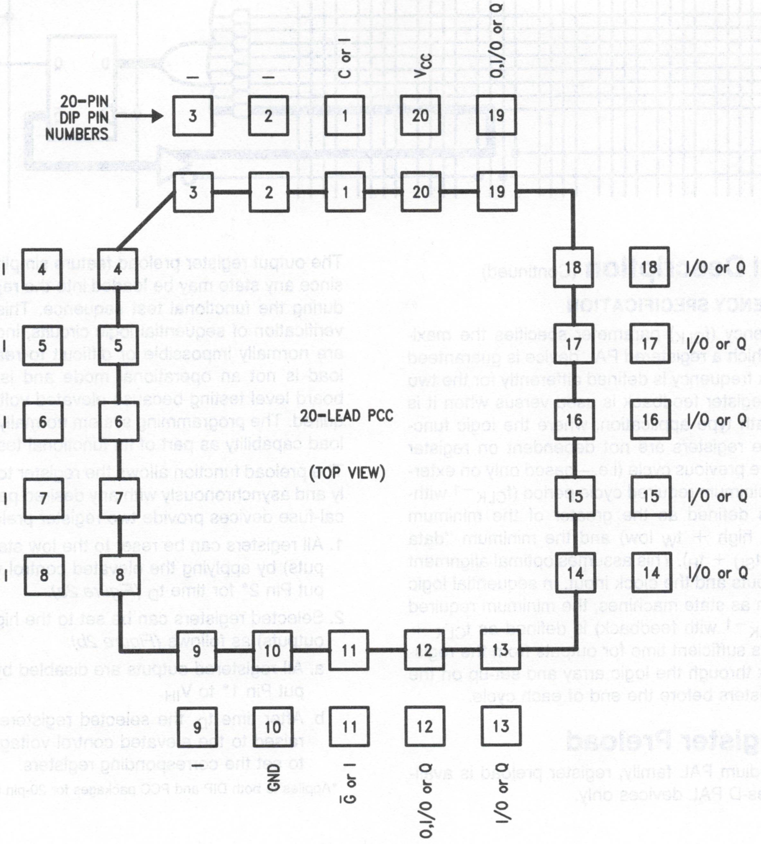
TL/L/9391-12

PAL16R8


C
I
VCC
Q
GND
G̅
AND ARRAY
8
D Q
Q̅
PCC PIN NUMBERS

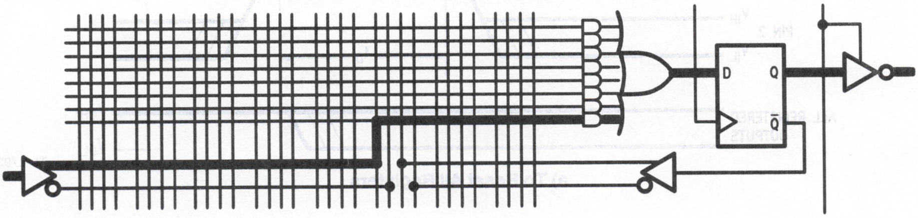
TL/L/9391-13

20-Lead PCC Connection Conversion Diagram


I
I
C or I
V_{CC}
0,I/O or Q
20-PIN DIP PIN NUMBERS
3
2
1
20
19
3
2
1
20
19
I 4
4
I 5
5
I 6
6
I 7
7
I 8
8
18
18 I/O or Q
17
17 I/O or Q
16
16 I/O or Q
15
15 I/O or Q
14
14 I/O or Q
20-LEAD PCC
(TOP VIEW)
9
10
11
12
13
9
10
11
12
13
GND
$\overline{G}$ or I
0,I/O or Q
I/O or Q

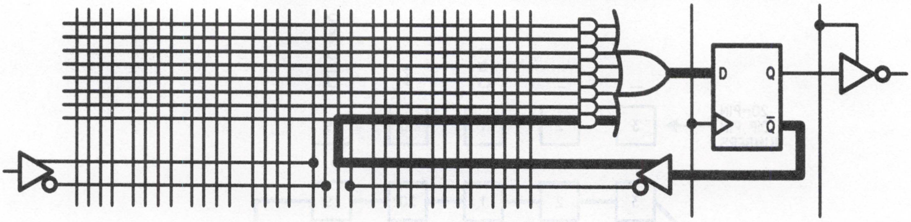
TL/L/9391–14

Typical Registered Logic Function Without Feedback


D
Q
$\overline{Q}$

TL/L/9391–15

Typical Registered Logic Function With Feedback



TL/L/9391-16

Functional Description (Continued)

CLOCK FREQUENCY SPECIFICATION

The clock frequency (f_{CLK}) parameter specifies the maximum speed at which a registered PAL device is guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, where the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e.—based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_W high + t_W low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period (f_{CLK}^{-1} with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set-up on the inputs to the registers before the end of each cycle.

Output Register Preload

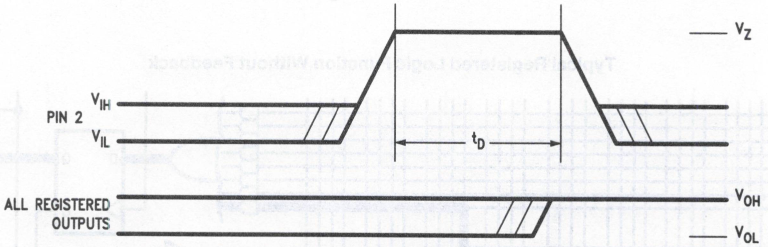
In the 20-pin Medium PAL family, register preload is available on the Series-D PAL devices only.

The output register preload feature simplifies device testing since any state may be loaded into the registers at any time during the functional test sequence. This allows complete verification of sequential logic circuits, including states that are normally impossible or difficult to reach. Register preload is not an operational mode and is not intended for board level testing because elevated voltage levels are required. The programming system normally provides the preload capability as part of its functional test facility.

The preload function allows the register to be loaded directly and asynchronously with any desired pattern. These vertical-fuse devices provide two register preload operations:

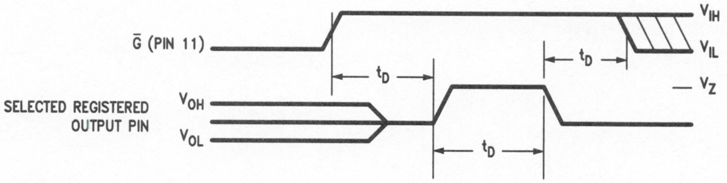
1. All registers can be reset to the low state (high-level outputs) by applying the elevated control voltage (V_Z) to input Pin 2* for time t_D *(Figure 2a)*.
2. Selected registers can be set to the high state (low-level outputs) as follows *(Figure 2b)*:
   a. All registered outputs are disabled by raising the $\overline{G}$ input Pin 1* to V_{IH}.
   b. After time t_D, the selected registered output pins are raised to the elevated control voltage (V_Z) for time t_D to set the corresponding registers.

*Applies to both DIP and PCC packages for 20-pin PAL devices.



TL/L/9391-26

a) To Reset All Registers



TL/L/9391-27

b) To Set Selected Registers

Note: V_Z = 9.5V to 10.0V, t_D min. = 500 ns.

FIGURE 2. Series-D Register Preload Waveforms

Security Fuse

Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses. Register preload is not affected by the security fuses.

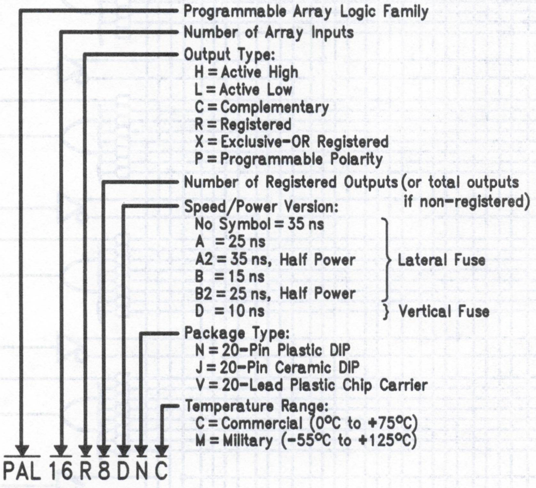
Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a variety of programming equipment. Many software packages and programming units support a wide variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

In National Series-D PAL devices, logical and physical connections between array input lines and product-term lines are established when vertical fuse cells are programmed. This is opposite to other PAL products based on fusible links in which connections are established when fuses are left unprogrammed (intact). This difference is compensated by the vertical-fuse PAL programming algorithm so that the *user's design development process looks the same.* (The only functional difference due to vertical-fuse technology is the behavior of "unprogrammed" devices.) The JEDEC programming maps produced by PAL development software for all Medium PAL devices denote a "connection" with a "0", and a "non-connection" with a "1". The programming algorithms for most fuse-link PLDs program fuses where ones are located in the map to remove corresponding connections, whereas the algorithm for National Series-D PAL products automatically compensates by programming vertical-fuse cells where zeroes are located in the map to establish connections. Therefore, the *same JEDEC map* representing the user's desired logic equations produces the *same functional results* when using either PAL technology. The user need only provide the appropriate device code and/or adapter for the programming equipment to invoke the proper programming algorithm. Only programmers with the certified National Series-D vertical-fuse PAL programming algorithm can be used to program these vertical-fuse devices.

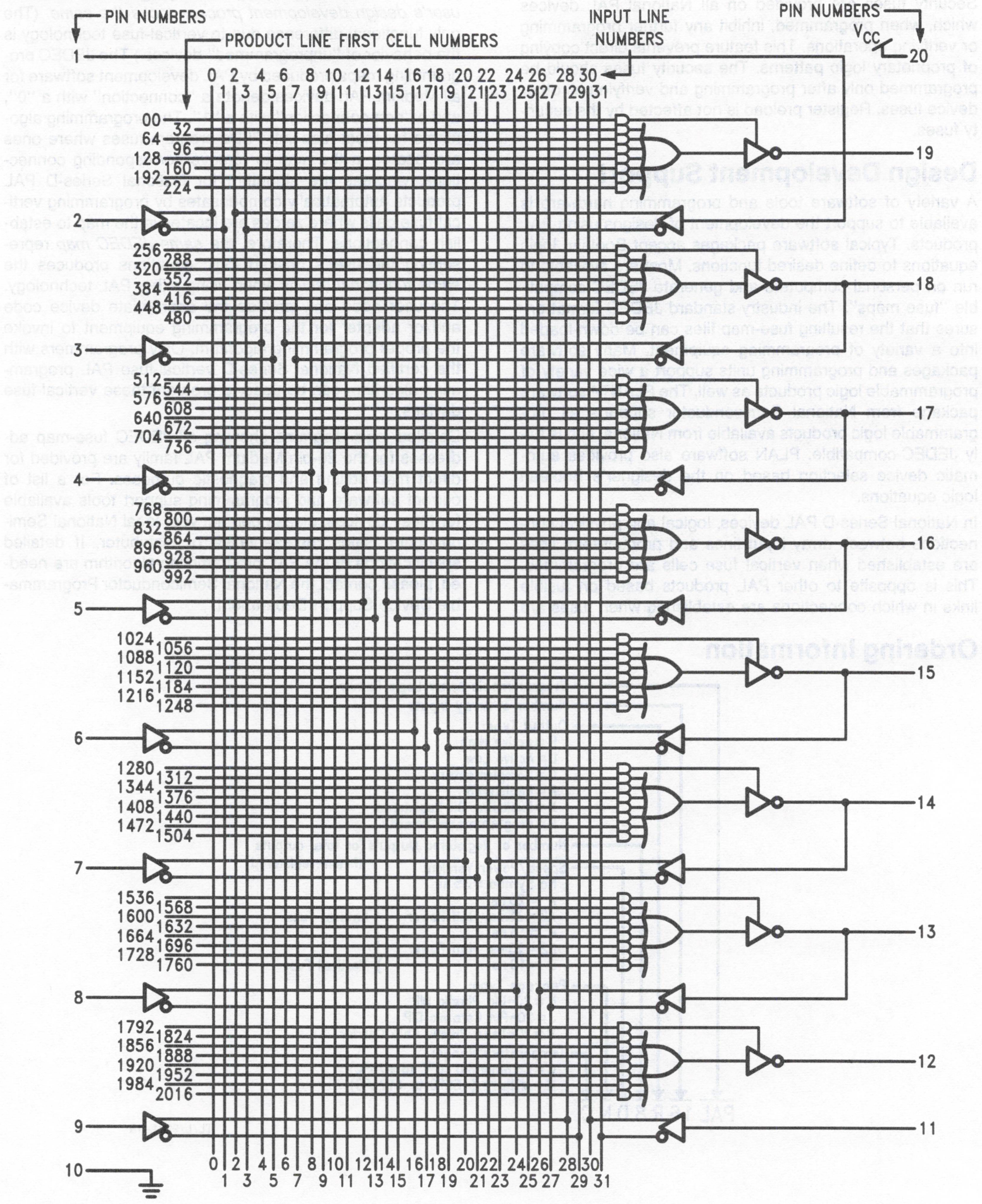
Detailed logic diagrams showing all JEDEC fuse-map addresses for the 20-pin Medium PAL family are provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

Ordering Information


Programmable Array Logic Family
Number of Array Inputs
Output Type:
H = Active High
L = Active Low
C = Complementary
R = Registered
X = Exclusive-OR Registered
P = Programmable Polarity
Number of Registered Outputs (or total outputs if non-registered)
Speed/Power Version:
No Symbol = 35 ns
A = 25 ns
A2 = 35 ns, Half Power
B = 15 ns
B2 = 25 ns, Half Power
Lateral Fuse
D = 10 ns
Vertical Fuse
Package Type:
N = 20-Pin Plastic DIP
J = 20-Pin Ceramic DIP
V = 20-Lead Plastic Chip Carrier
Temperature Range:
C = Commercial (0°C to +75°C)
M = Military (−55°C to +125°C)
PAL 16 R 8 D N C

TL/L/9391-18

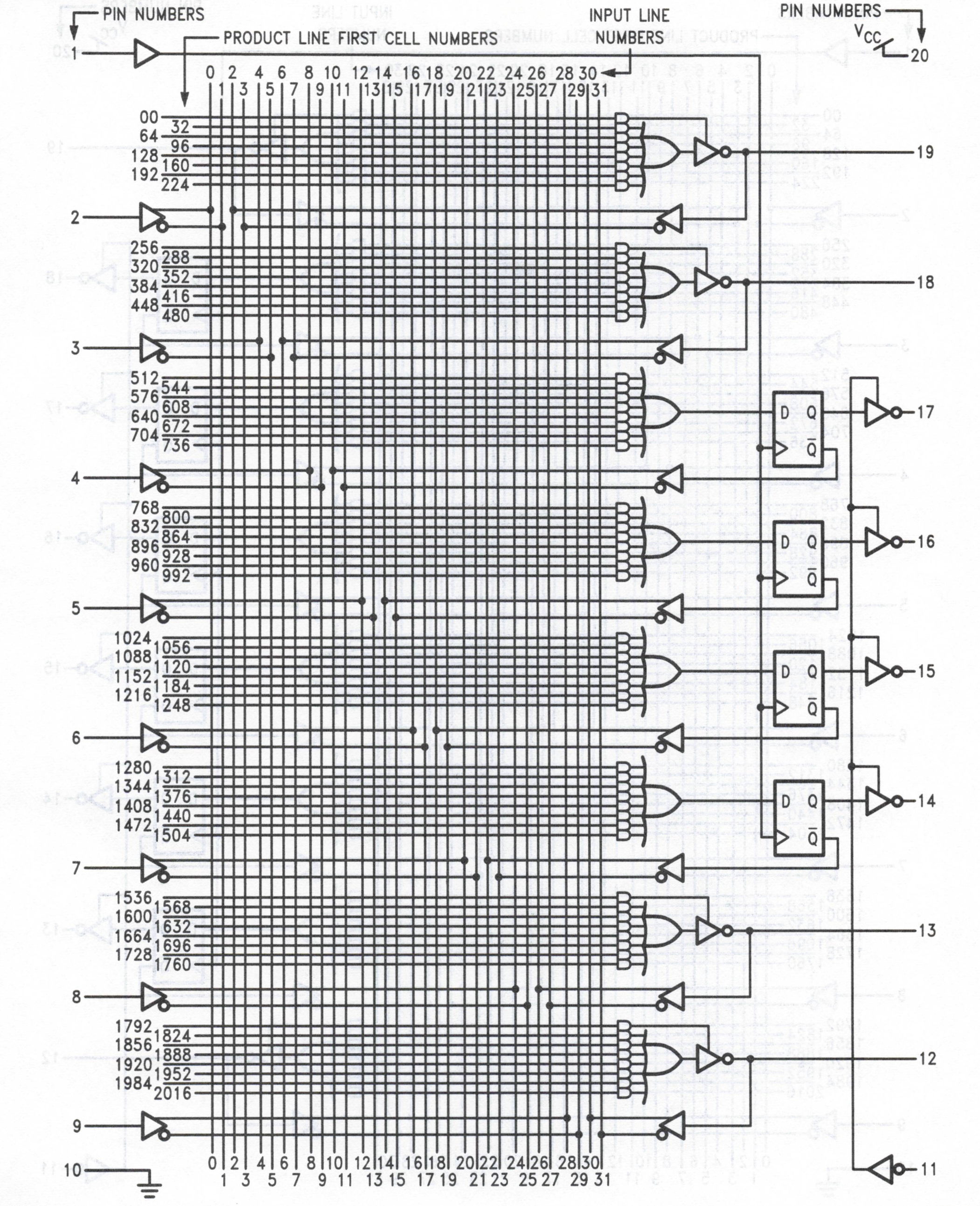
Logic Diagram—PAL16L8


PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
PIN NUMBERS
VCC
20
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
00
32
64
96
128
160
192
224
256
288
320
352
384
416
448
480
512
544
576
608
640
672
704
736
768
800
832
864
896
928
960
992
1024
1056
1088
1120
1152
1184
1216
1248
1280
1312
1344
1376
1408
1440
1472
1504
1536
1568
1600
1632
1664
1696
1728
1760
1792
1824
1856
1888
1920
1952
1984
2016

TL/L/9391-19

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

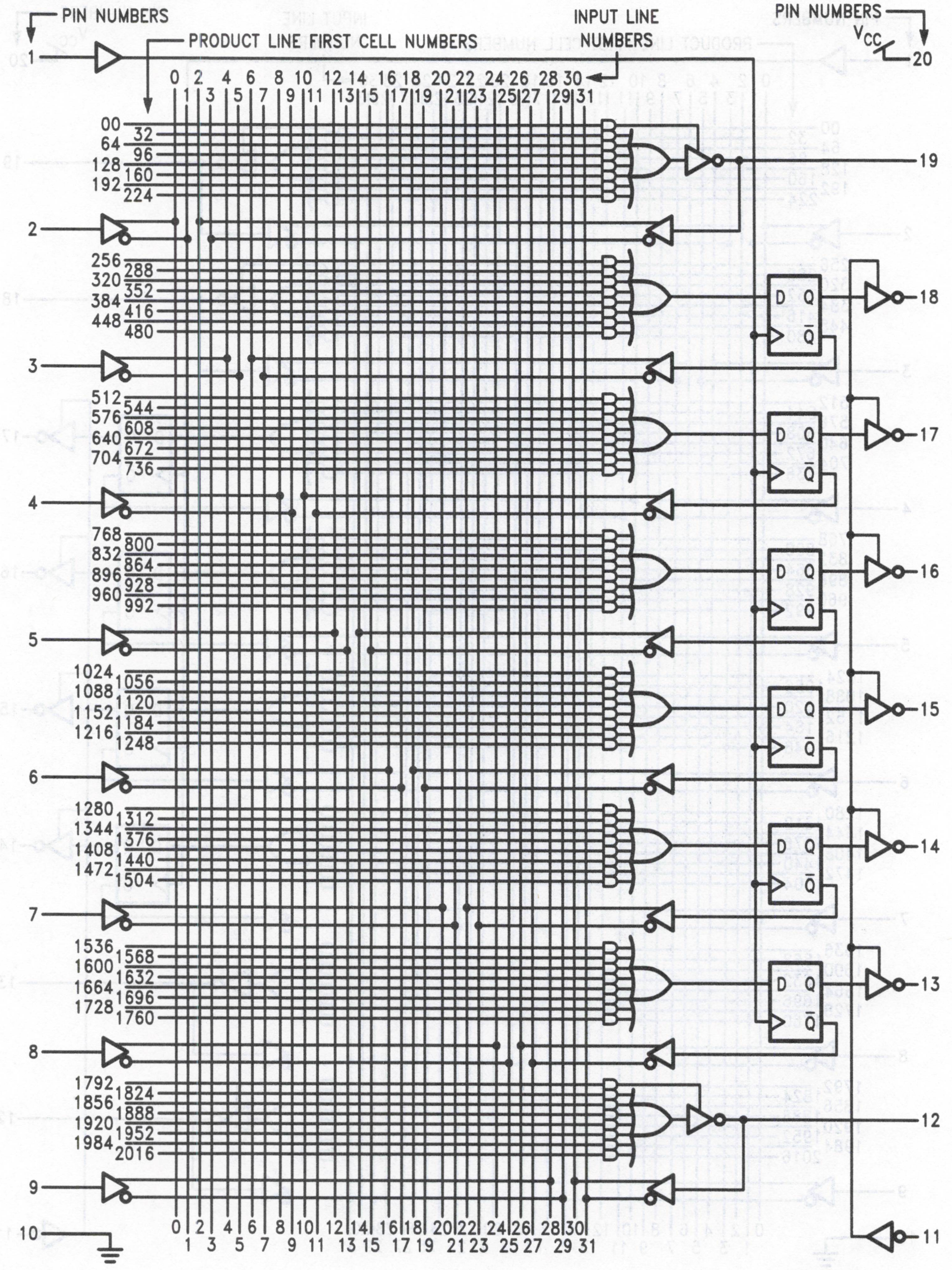
Logic Diagram—PAL16R4


PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
PIN NUMBERS
V_{CC}
20
1
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
00 32 64 96 128 160 192 224
19
2
256 288 320 352 384 416 448 480
18
3
512 544 576 608 640 672 704 736
D Q Q̄
17
4
768 800 832 864 896 928 960 992
D Q Q̄
16
5
1024 1056 1088 1120 1152 1184 1216 1248
D Q Q̄
15
6
1280 1312 1344 1376 1408 1440 1472 1504
D Q Q̄
14
7
1536 1568 1600 1632 1664 1696 1728 1760
13
8
1792 1824 1856 1888 1920 1952 1984 2016
12
9
10
11

TL/L/9391–20

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

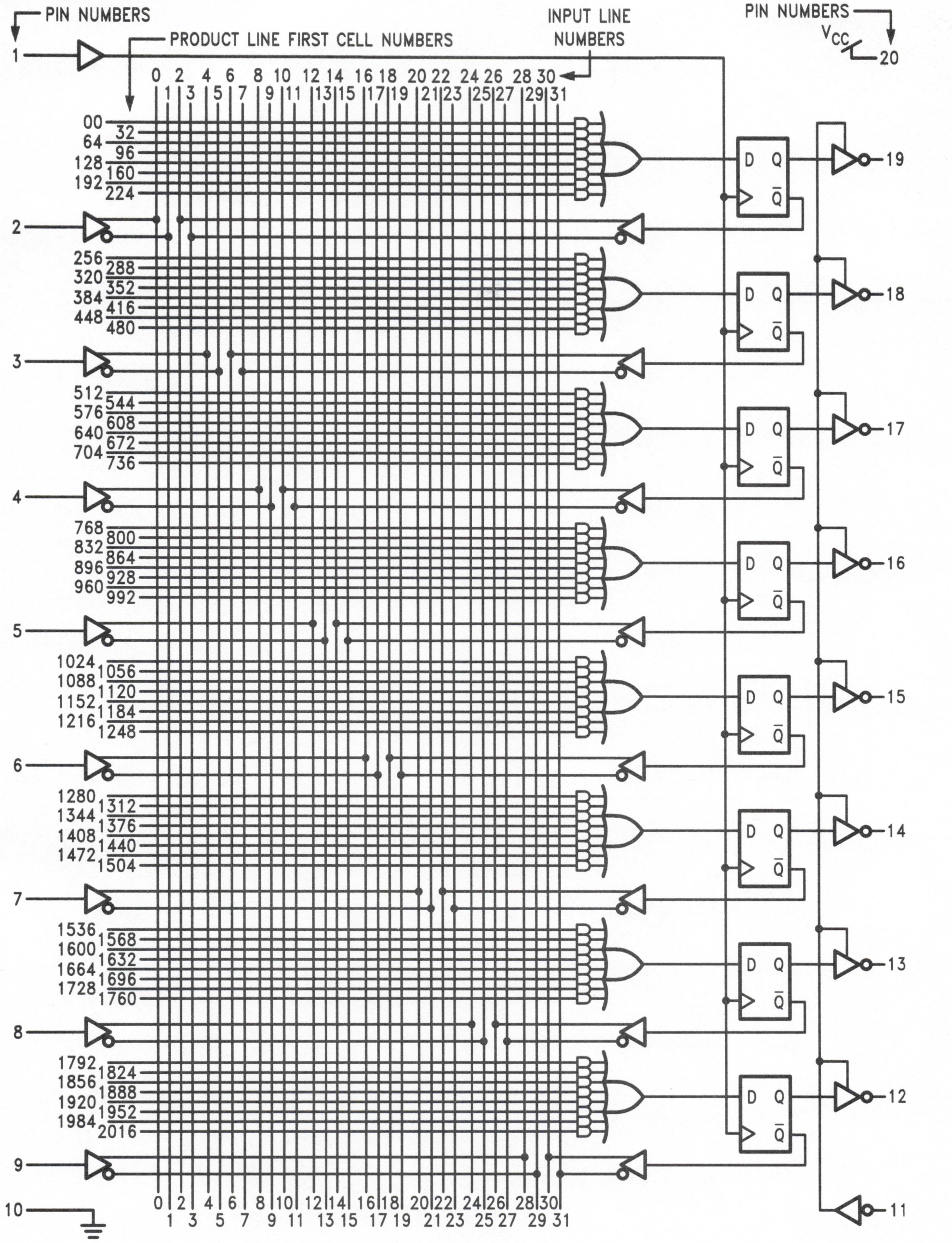
Logic Diagram—PAL16R6


PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
PIN NUMBERS
VCC
20
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
1
00 32 64 96 128 160 192 224
19
2
256 288 320 352 384 416 448 480
D Q
Q̄
18
3
512 544 576 608 640 672 704 736
17
4
768 800 832 864 896 928 960 992
16
5
1024 1056 1088 1120 1152 1184 1216 1248
15
6
1280 1312 1344 1376 1408 1440 1472 1504
14
7
1536 1568 1600 1632 1664 1696 1728 1760
13
8
1792 1824 1856 1888 1920 1952 1984 2016
12
9
10
11

TL/L/9391-21

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

Logic Diagram—PAL16R8


PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
PIN NUMBERS
V_{CC}
20
1
2
3
4
5
6
7
8
9
10
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
00 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
D Q
$\overline{Q}$
19
18
17
16
15
14
13
12
11

TL/L/9391–22

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

National Semiconductor

Programmable Array Logic (PAL®) 24-Pin Small PAL Family

General Description

The 24-pin Small PAL family contains six popular PAL architectures. The devices in the Small PAL family draw only 100 mA maximum supply current as compared to 210 mA in the 24-pin Medium PAL devices. These devices offer speeds as fast as 25 ns maximum propagation delay. National Semiconductor's Schottky TTL process with titanium tungsten fusible links provides high-speed user-programmable replacements for conventional SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming fusible links to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

The Small PAL logic array has between 12 and 20 complementary inputs and up to 10 combinatorial outputs generated by a single programmable AND-gate array with fixed OR-gate connections. The Small PAL family offers a variety of input/output combinations as shown in the Device Types table below. Security fuses can be programmed to prevent direct copying of proprietary logic patterns.

Features

- As fast as 25 ns maximum propagation delay
- User-programmable replacement for TTL logic
- Large variety of JEDEC-compatible programming equipment available
- Fully supported by National PLAN™ development software
- Security fuse prevents direct copying of logic patterns

Device Types

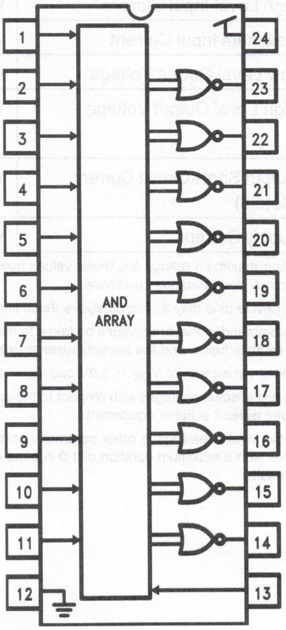
Device Type	Dedicated Inputs	Combinatorial Outputs
PAL12L10	12	10
PAL14L8	14	8
PAL16L6	16	6
PAL18L4	18	4
PAL20L2	20	2
PAL20C1	20	1 Pair

Speed/Power Versions

Series	Example	Commercial		Military	
		t_{PD}	I_{CC}	t_{PD}	I_{CC}
Standard	PAL12L10	40 ns	100 mA	45 ns	100 mA
A	PAL12L10A	25 ns*	100 mA	30 ns*	100 mA

*Except PAL20C1A t_{PD} = 30 ns Commercial, 35 ns Military.

Block Diagram—PAL12L10



TL/L/9997-1

Standard Series (PAL12L10, PAL14L8, PAL16L6, PAL18L4, PAL20L2, PAL20C1)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	-0.5 to $+7.0$V
Input Voltage (Notes 2 and 3)	-1.5 to $+5.5$V
Off-State Output Voltage (Note 2)	-1.5 to $+5.5$V
Input Current (Note 2)	-30.0 mA to $+5.0$ mA
Output Current (I_{OL})	$+100$ mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	Military			Commercial			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature	−55			0		75	°C
T_C	Operating Case Temperature			125				°C

Electrical Characteristics Over Recommended Operating Conditions (Note 4)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 5)						0.8	V
V_{IH}	High Level Input Voltage (Note 5)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, $I = -18$ mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current	V_{CC} = Max, $V_I = 0.4$V				−0.02	−0.25	mA
I_{IH}	High Level Input Current	V_{CC} = Max, $V_I = 2.4$V					25	µA
I_I	Maximum Input Current	V_{CC} = Max, $V_I = 5.5$V					1	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	$I_{OL} = 8$ mA			0.3	0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	$I_{OH} = -2$ mA	MIL	2.4	3.0		V
			$I_{OH} = -3.2$ mA	COM				
I_{OS}	Output Short-Circuit Current (Note 6)	$V_{CC} = 5$V, $V_O = 0$V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				60	100	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 (DIP) are connected directly to the security fuses, and the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: All typical values are for $V_{CC} = 5.0$V and $T_A = 25$°C.

Note 5: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

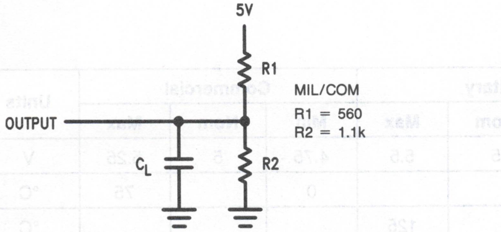
Note 6: To avoid invalid readings in other parameter tests, it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 second each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Standard Series (PAL12L10, PAL14L8, PAL16L6, PAL18L4, PAL20L2, PAL20C1) (Continued)

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input to Output	$C_L = 50$ pF		25	45		25	40	ns

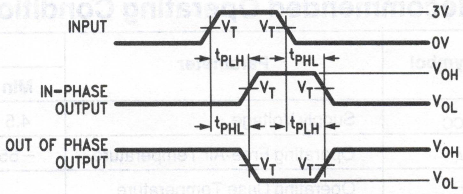
Test Load


5V
R1
OUTPUT
C_L
R2
MIL/COM
R1 = 560
R2 = 1.1k

TL/L/9997-2

Test Waveform

Propagation Delay


INPUT
3V
0V
V_T
V_T
t_{PLH}
t_{PHL}
IN-PHASE OUTPUT
V_{OH}
V_{OL}
V_T
V_T
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT
V_{OH}
V_{OL}
V_T
V_T

TL/L/9997-3

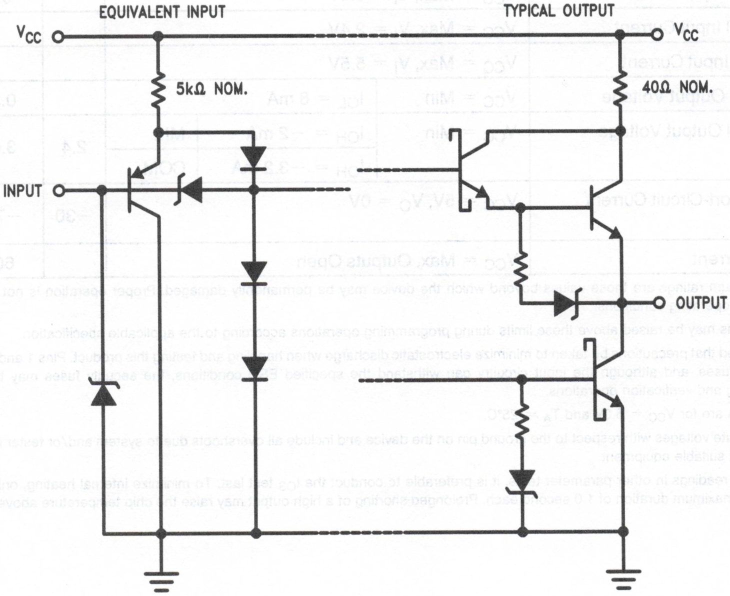
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Schematic of Inputs and Outputs


EQUIVALENT INPUT
TYPICAL OUTPUT
V_{CC}
V_{CC}
5kΩ NOM.
40Ω NOM.
INPUT
OUTPUT

TL/L/9997-4

Series A (PAL12L10A, PAL14L8A, PAL16L6A, PAL18L4A, PAL20L2A, PAL20C1A)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5 to +7.0V
Input Voltage (Note 2)	−1.5 to +5.5V
Off-State Output Voltage (Note 2)	−1.5 to +5.5V
Input Current (Note 2)	−30.0 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance (Note 3)	1500V

$C_{ZAP} = 100$ pF
$R_{ZAP} = 1500\Omega$
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter	Military			Commercial			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature	−55			0		75	°C
T_C	Operating Case Temperature			125				°C

Electrical Characteristics Over Recommended Operating Conditions (Note 4)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 5)						0.8	V
V_{IH}	High Level Input Voltage (Note 5)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 8 mA			0.3	0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	3.0		V
			I_{OH} = −3.2 mA	COM				
I_{OS}	Output Short-Circuit Current (Note 6)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				60	100	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 (DIP) are connected directly to the security fuses, and although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: All typical values are for V_{CC} = 5.0V and T_A = 25°C.

Note 5: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

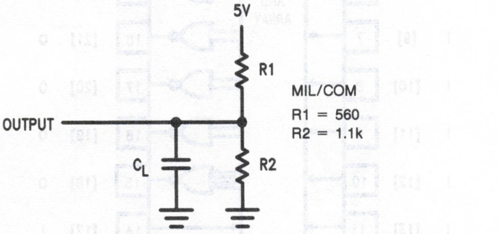
Note 6: To avoid Invalid readings in other parameter tests, it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 second each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Series A (PAL12L10A, PAL14L8A, PAL16L6A, PAL18L4A, PAL20L2A, PAL20C1A) (Continued)

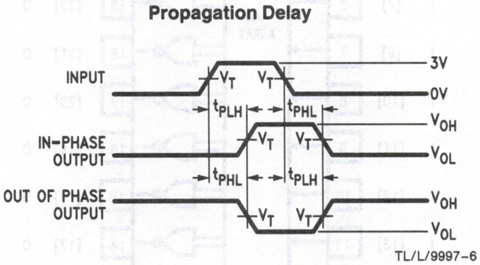
Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions		Military			Commercial			Units
				Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input to Output	C_L = 50 pF	12L10A, 14L8A, 16L6A, 18L4A, 20L2A		15	30		15	25	ns
			20C1A			35			30	ns

Test Load


5V
R1
OUTPUT
C_L
R2
MIL/COM
R1 = 560
R2 = 1.1k
TL/L/9997-5

Test Waveform


Propagation Delay
INPUT
IN-PHASE OUTPUT
OUT OF PHASE OUTPUT
3V
0V
V_{OH}
V_{OL}
V_{OH}
V_{OL}
V_T
t_{PLH}
t_{PHL}
t_{PHL}
t_{PLH}
TL/L/9997-6

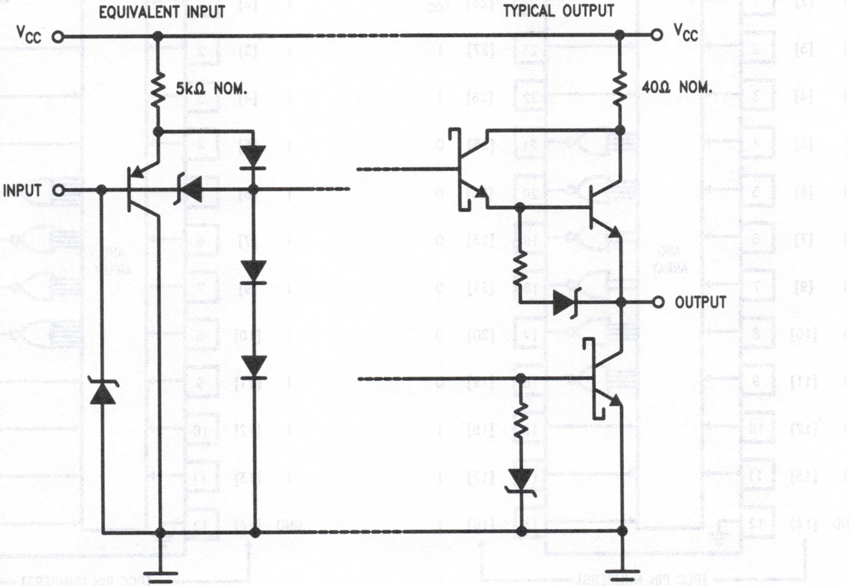
Notes:

V_T = 1.5V

C_L includes probe and jig capacitance.

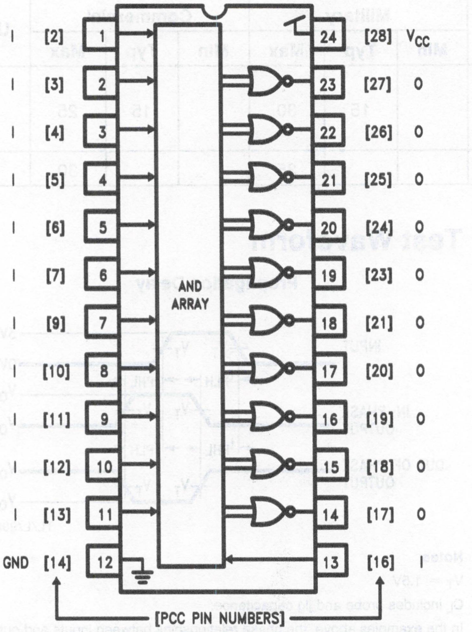
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Schematic of Inputs and Outputs


EQUIVALENT INPUT
TYPICAL OUTPUT
V_{CC}
V_{CC}
5kΩ NOM.
40Ω NOM.
INPUT
OUTPUT
TL/L/9997-7

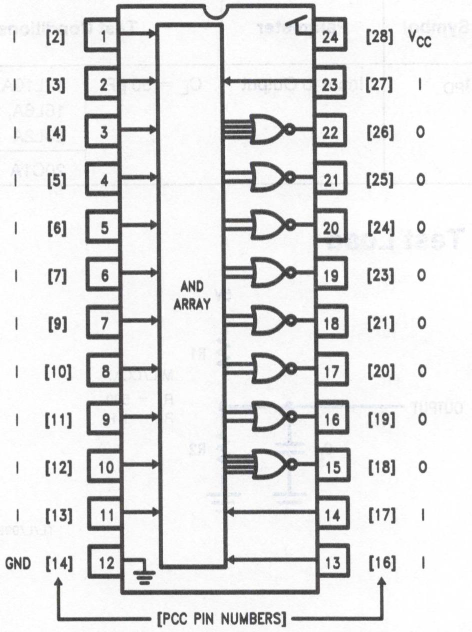
24-Pin Small PAL Family Block Diagrams—DIP Connections

PAL12L10


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] O
22 [26] O
21 [25] O
20 [24] O
19 [23] O
18 [21] O
17 [20] O
16 [19] O
15 [18] O
14 [17] O
13 [16] I
AND ARRAY
[PCC PIN NUMBERS]

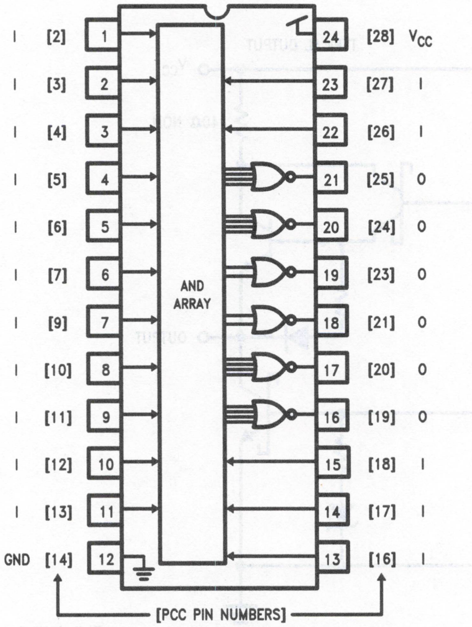
TL/L/9997-8

PAL14L8


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] I
22 [26] O
21 [25] O
20 [24] O
19 [23] O
18 [21] O
17 [20] O
16 [19] O
15 [18] O
14 [17] I
13 [16] I
AND ARRAY
[PCC PIN NUMBERS]

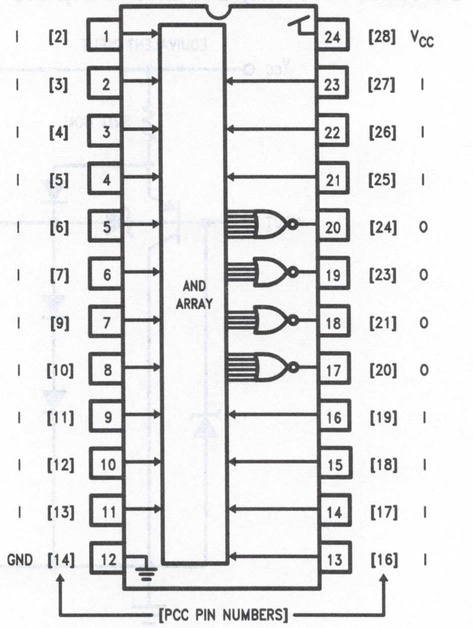
TL/L/9997-9

PAL16L6


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] I
22 [26] I
21 [25] O
20 [24] O
19 [23] O
18 [21] O
17 [20] O
16 [19] O
15 [18] I
14 [17] I
13 [16] I
AND ARRAY
[PCC PIN NUMBERS]

TL/L/9997-10

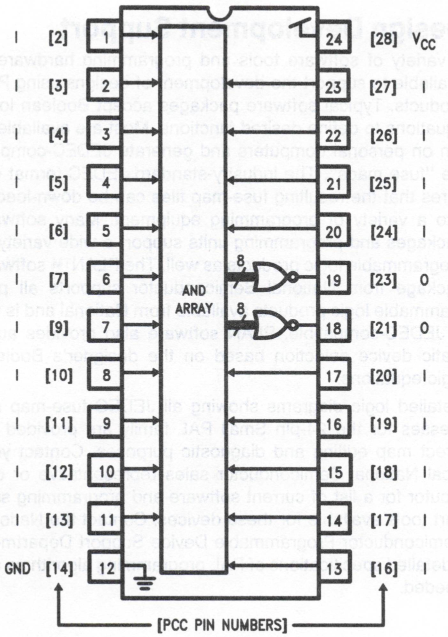
PAL18L4


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] I
22 [26] I
21 [25] I
20 [24] O
19 [23] O
18 [21] O
17 [20] O
16 [19] I
15 [18] I
14 [17] I
13 [16] I
AND ARRAY
[PCC PIN NUMBERS]

TL/L/9997-11

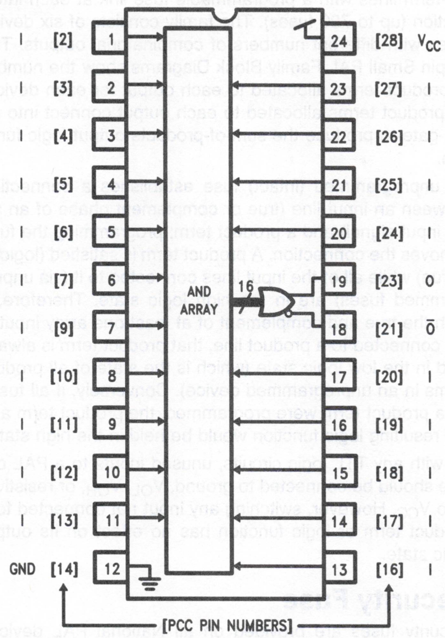
24-Pin Small PAL Family Block Diagrams—DIP Connections (Continued)

PAL20L2


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] V_{CC}
23 [27] I
22 [26] I
21 [25] I
20 [24] I
19 [23] O
18 [21] O
17 [20] I
16 [19] I
15 [18] I
14 [17] I
13 [16] I
AND ARRAY
8
8
[PCC PIN NUMBERS]

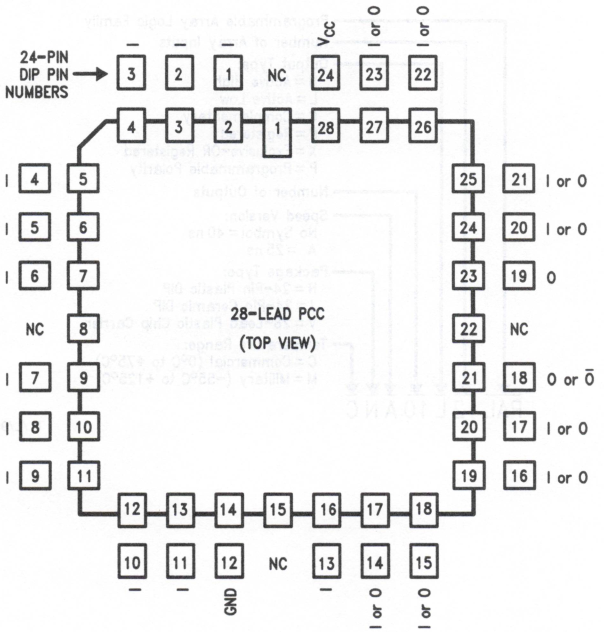
TL/L/9997-12

PAL20C1


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] V_{CC}
23 [27] I
22 [26] I
21 [25] I
20 [24] I
19 [23] O
18 [21] $\overline{O}$
17 [20] I
16 [19] I
15 [18] I
14 [17] I
13 [16] I
AND ARRAY
16
[PCC PIN NUMBERS]

TL/L/9997-13

28-Lead PCC Connection Conversion Diagram


24-PIN DIP PIN NUMBERS
I 3
I 2
I 1
NC
V_{CC} 24
I or O 23
I or O 22
4 3 2 1 28 27 26
I 4 5
I 5 6
I 6 7
NC 8
I 7 9
I 8 10
I 9 11
25 21 I or O
24 20 I or O
23 19 O
22 NC
21 18 O or $\overline{O}$
20 17 I or O
19 16 I or O
12 13 14 15 16 17 18
10 I
11 I
12 GND
NC
13 I
14 I or O
15 I or O
28-LEAD PCC (TOP VIEW)

TL/L/9997-14

Note: For availability of old (NON-JEDEC) pinout, please contact your local National Semiconductor sales representative or distributor.

Functional Description

The 24-pin Small PAL logic arrays consist of between 12 and 20 complementary input lines and either 16 or 20 product-term lines with a programmable fuse link at each intersection (up to 720 fuses). The family consists of six device types with different numbers of combinatorial outputs. The 24-pin Small PAL Family Block Diagrams show the number of product terms allocated to each output for each device. All product terms allocated to each output connect into an OR-gate to produce the sum-of-products output logic function.

An unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses) are in the high logic state. Therefore, if both the true and complement of at least one array input is left connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed device). Conversely, if all fuses on a product term were programmed, the product term and the resulting logic function would be held in the high state.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

Security Fuse

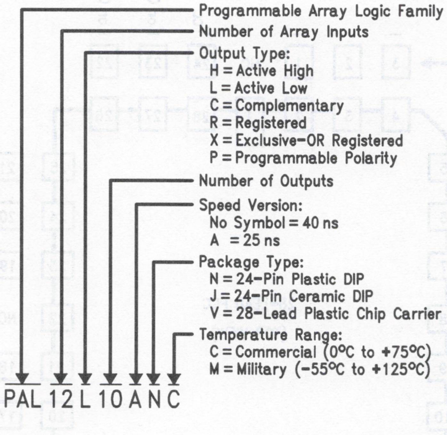
Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses.

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a variety of programming equipment. Many software packages and programming units support a wide variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

Detailed logic diagrams showing all JEDEC fuse-map addresses for the 24-pin Small PAL family are provided for direct map editing and diagnostic purposes. Contact your local National Semiconductor sales representative or distributor for a list of current software and programming support tools available for these devices. Contact the National Semiconductor Programmable Device Support Department if detailed specifications of PAL programming algorithms are needed.

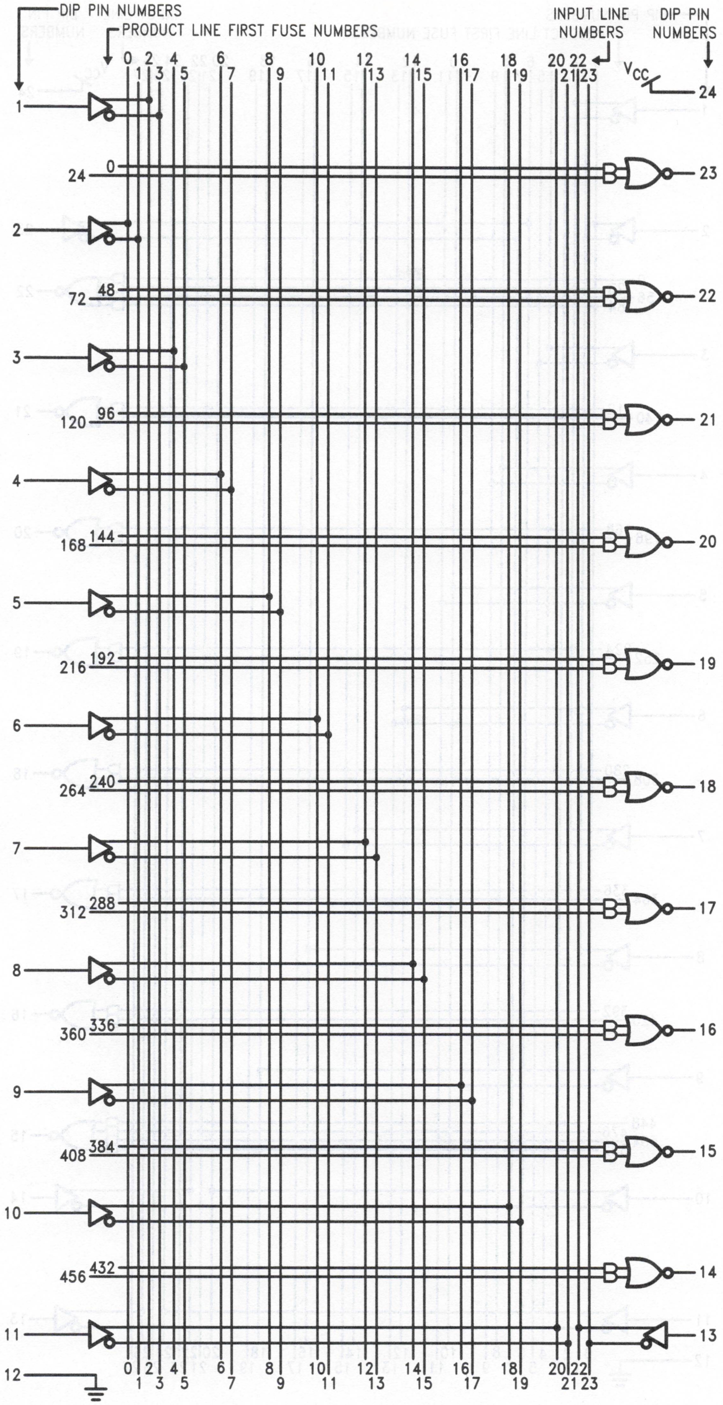
Ordering Information



Programmable Array Logic Family
Number of Array Inputs
Output Type:
H = Active High
L = Active Low
C = Complementary
R = Registered
X = Exclusive-OR Registered
P = Programmable Polarity
Number of Outputs
Speed Version:
No Symbol = 40 ns
A = 25 ns
Package Type:
N = 24-Pin Plastic DIP
J = 24-Pin Ceramic DIP
V = 28-Lead Plastic Chip Carrier
Temperature Range:
C = Commercial (0°C to +75°C)
M = Military (−55°C to +125°C)
PAL 12 L 10 A N C

TL/L/9997-15

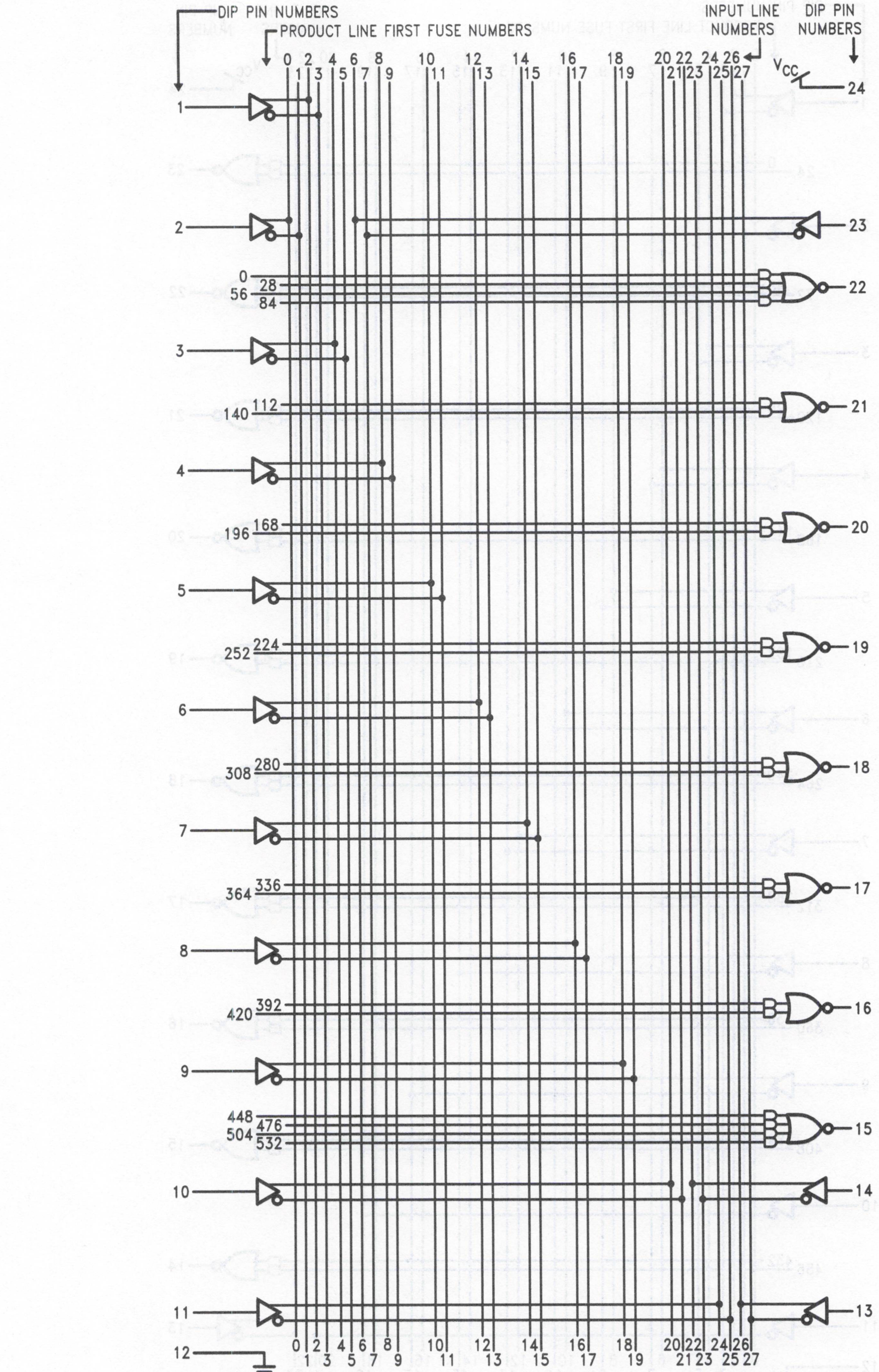
Logic Diagram PAL12L10


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
VCC
24
1
0
24
23
2
48
72
22
3
96
120
21
4
144
168
20
5
192
216
19
6
240
264
18
7
288
312
17
8
336
360
16
9
384
408
15
10
432
456
14
11
13
12

TL/L/9997-16

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

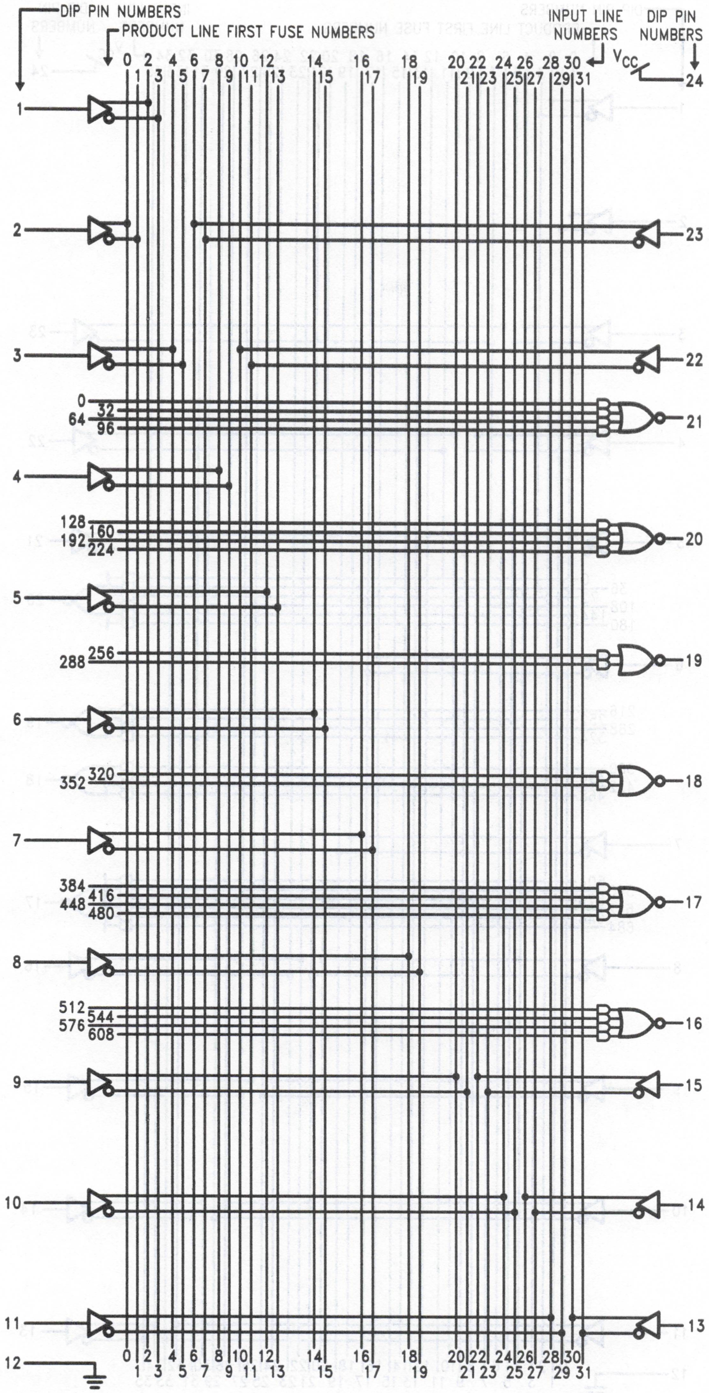
Logic Diagram PAL14L8


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
0 2 4 6 8 10 12 14 16 18 20 22 24 26
1 3 5 7 9 11 13 15 17 19 21 23 25 27
VCC
24
1
2
23
0
28
56
84
22
3
112
140
21
4
168
196
20
5
224
252
19
6
280
308
18
7
336
364
17
8
392
420
16
9
448
476
504
532
15
10
14
11
13
12

TL/L/9997-17

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

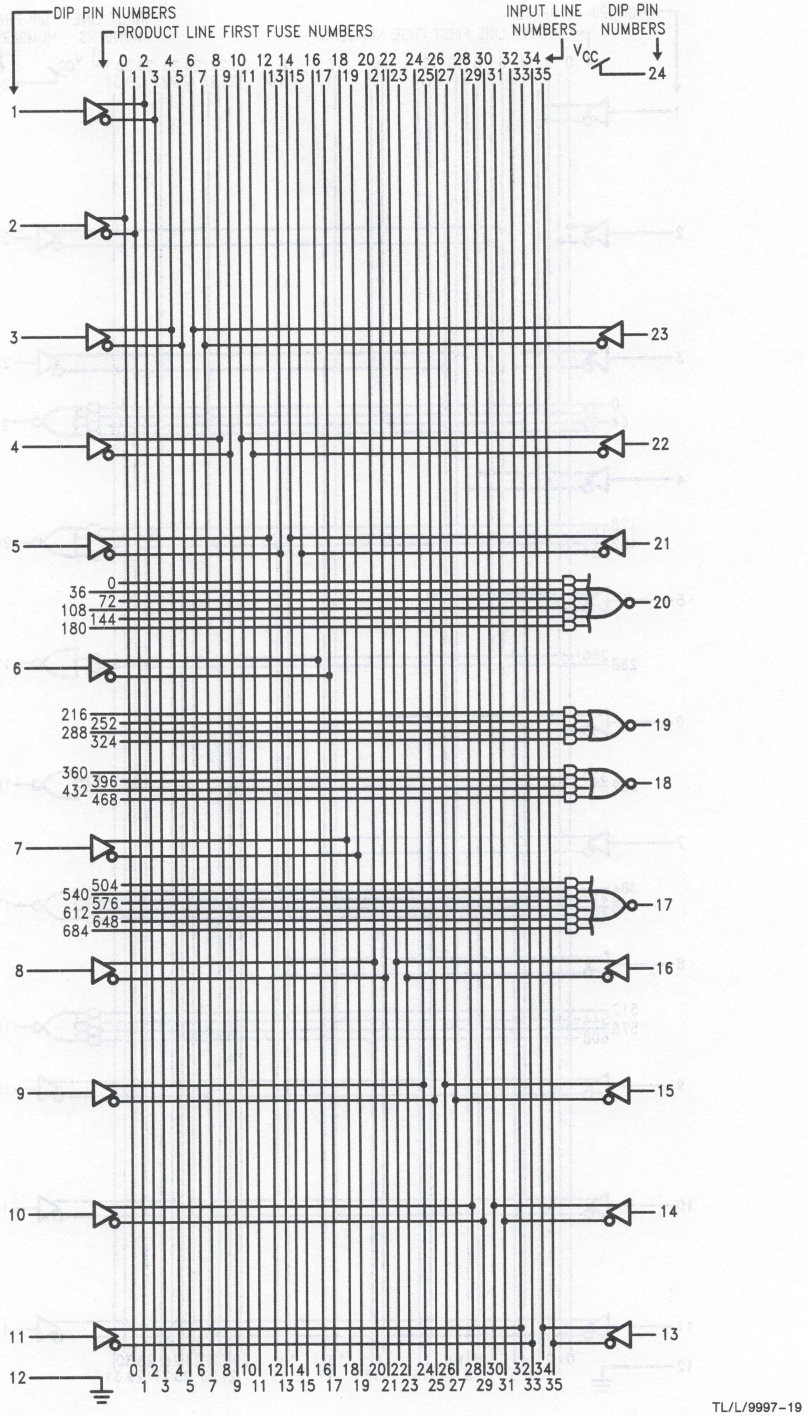
Logic Diagram PAL16L6


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
0
32
64
96
128
160
192
224
256
288
320
352
384
416
448
480
512
544
576
608

TL/L/9997-18

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

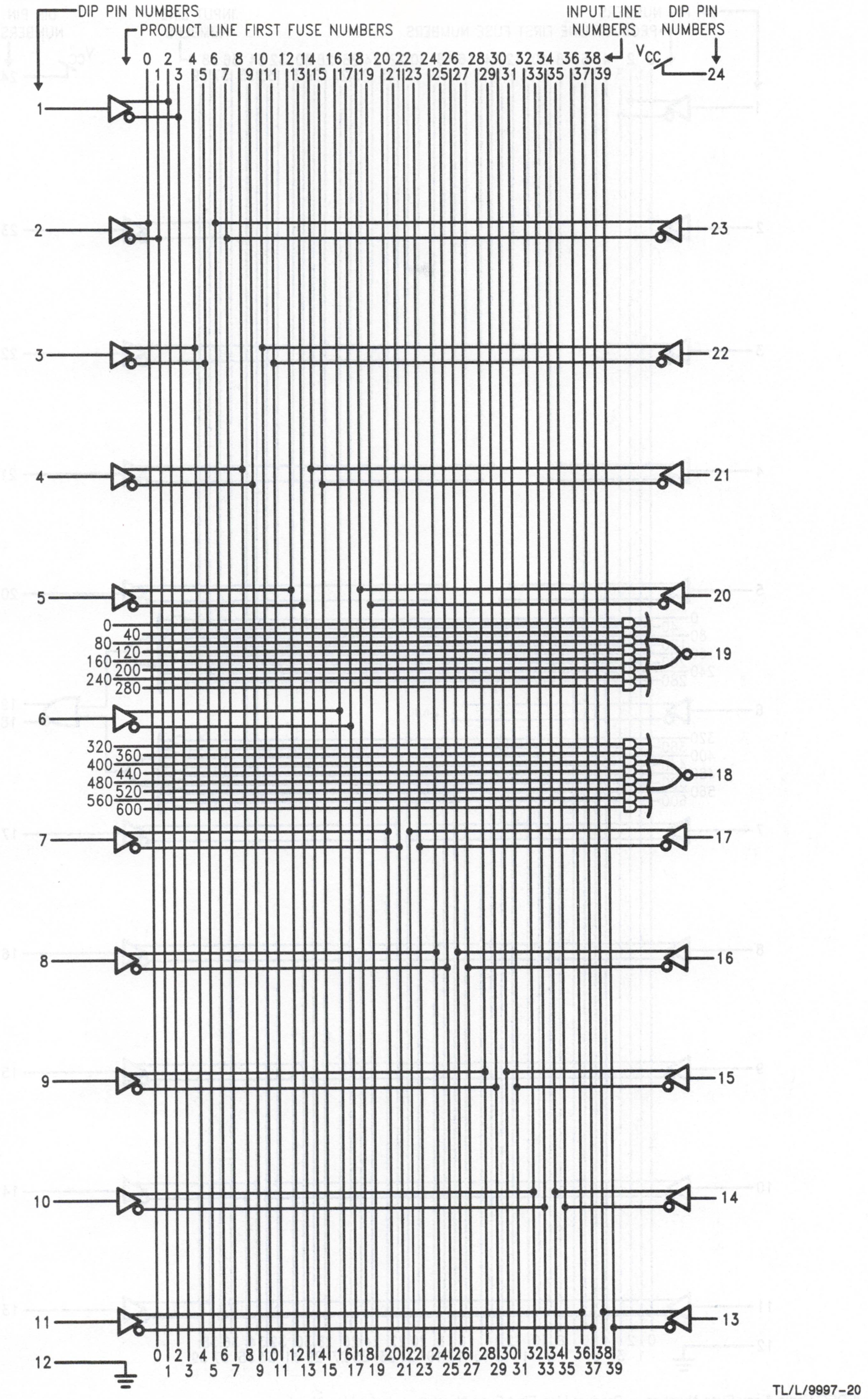
Logic Diagram PAL18L4


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
0
36
72
108
144
180
216
252
288
324
360
396
432
468
504
540
576
612
648
684

TL/L/9997-19

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

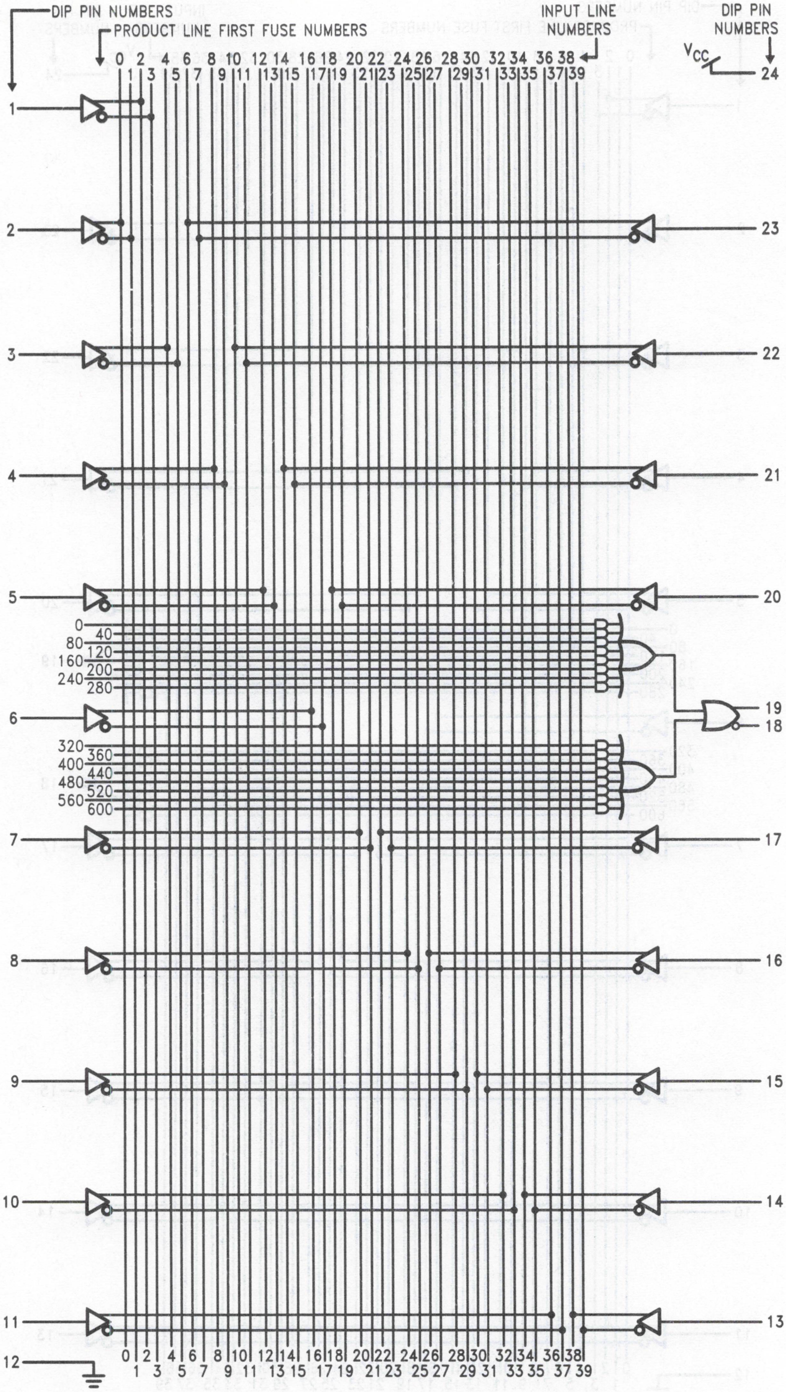
Logic Diagram PAL20L2


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
24
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
2
23
3
22
4
21
5
20
0
40
80
120
160
200
240
280
19
6
320
360
400
440
480
520
560
600
18
7
17
8
16
9
15
10
14
11
13
12

TL/L/9997-20

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

Logic Diagram PAL20C1


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
24
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
2
3
4
5
6
7
8
9
10
11
12
23
22
21
20
19
18
17
16
15
14
13
0
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600

TL/L/9997-21

Note: JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

National Semiconductor

Programmable Array Logic (PAL®) 24-Pin Exclusive-OR PAL Family

General Description

The 24-pin Exclusive-OR PAL family contains four industry-standard PAL architectures optimized for a specific class of applications. National Semiconductor's advanced Schottky TTL process with titanium tungsten fusible links provides high-speed user-programmable replacements for conventional SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming fusible links to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

The PAL logic array has a total of 20 complementary inputs and 10 outputs generated by a single programmable AND-gate array with fixed OR-gate connections. Device outputs are either taken directly from the AND-OR functions (combinatorial) or passed through exclusive-OR gates and D-type flip-flops (registered). Registers allow the PAL device to implement sequential logic circuits. The exclusive-OR functions provide easy implementation of the "hold" operation used in counters and other state sequences. TRI-STATE® outputs facilitate busing and provide bidirectional I/O capability. The exclusive-OR PAL family offers a variety of combinatorial and registered output mixtures, as shown in the Device Types table below.

Series-A devices have power-up reset and register preload features available. On power-up, all registers are reset to simplify sequential circuit design and testing. Direct register preload is also provided to facilitate device testing. Security fuses can be programmed to prevent direct copying of proprietary logic patterns.

Features

- As fast as 30 ns maximum propagation delay (combinatorial)
- Exclusive-OR function facilitates design of counters and state sequences
- User-programmable replacement for TTL logic
- Large variety of JEDEC-compatible programming equipment and design development software available
- Fully supported by National PLAN™ development software
- Power-up reset for registered outputs (Series-A)
- Register preload facilitates device testing (Series-A)
- Security fuse prevents direct copying of logic patterns

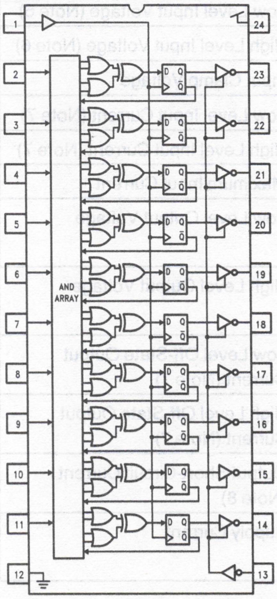
Device Types

Device Type	Dedicated Inputs	Registered Outputs (With Feedback)	Combinatorial	
			I/Os	Outputs
PAL20L10	14	—	6	2
PAL20X4	12	4	4	—
PAL20X8	12	6	2	
PAL20X10	12	8	—	—

Speed/Power Versions

Series	Example	Commercial		Military	
		t_{PD}	I_{CC}	t_{PD}	I_{CC}
Standard	PAL20L10	50 ns	165 mA	60 ns	165 mA
A	PAL20L10A	30 ns	165 mA	35 ns	165 mA

Block Diagram—PAL20X10



TL/L/9998-1

Standard Series (PAL20L10, PAL20X4, PAL20X8, PAL20X10)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Notes 2 and 3)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	40	20		35	20		ns
		High	30	10		25	10		ns
t_{SU}	Setup Time from Input or Feedback to Clock		60	38		50	38		ns
t_H	Hold Time of Input after Clock		0	−15		0	−15		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		17.2	10.0		17.2	12.5	MHz
		Without Feedback		33.3	14.3		33.3	16.7	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max,	20L10			90	165	mA
		Outputs Open	20X4, 20X8, 20X10			120	180	

Standard Series (PAL20L10, PAL20X4, PAL20X8, PAL20X10) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 (DIP) are connected directly to the security fuses, and the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(t_{WLOW} + t_{WHIGH})^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

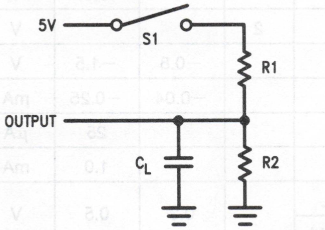
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commerical			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		35	60		35	50	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		20	40		20	30	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		20	45		20	35	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, from V_{OH}: S1 Open, from V_{OL}: S1 Closed		20	45		20	35	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		35	55		35	45	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, from V_{OH}: S1 Open, from V_{OL}: S1 Closed		35	55		35	45	ns

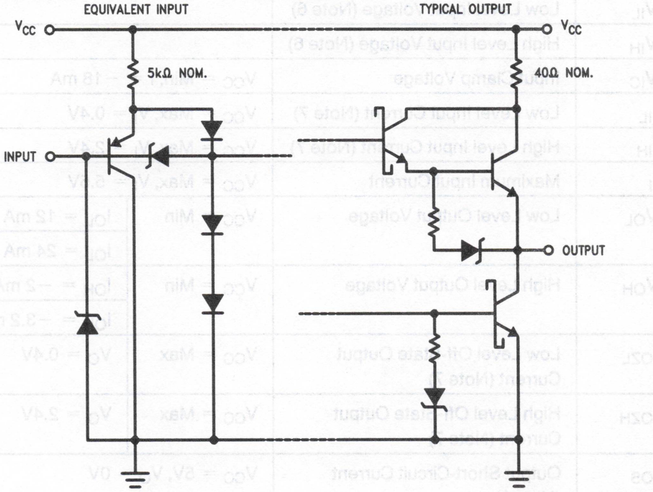
Test Load



TL/L/9998-2

MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

Schematic of Inputs and Outputs



TL/L/9998-3

Series—A (PAL20L10A, PAL20X4A, PAL20X8A, PAL20X10A)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30 mA to +5.0 mA
Output Current (I_{OL})	+100 mA

Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance (Note 3)	2000V

$C_{ZAP} = 100$ pF
$R_{ZAP} = 1500\Omega$
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	35	15		25	15		ns
		High	20	7		15	7		ns
t_{SU}	Setup Time from Input or Feedback to Clock		40	20		30	20		ns
t_H	Hold Time of Input after Clock		0	−15		0	−15		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		33.3	15.4		33.3	22.2	MHz
		Without Feedback		45.4	18.2		45.4	25.0	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.04	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	2.9		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open	20L10A			115	165	mA
			20X4A, 20X8A, 20X10A			135	180	

Series—A (PAL20L10A, PAL20X4A, PAL20X8A, PAL20X10A) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 (DIP) are connected directly to the security fuses, and although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(t_{WLOW} + t_{WHIGH})^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

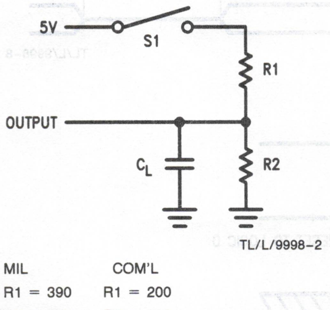
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commerical			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		23	35		23	30	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		10	25		10	15	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		11	25		11	20	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, from V_{OH}: S1 Open, from V_{OL}: S1 Closed		10	25		10	20	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		19	35		19	30	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, from V_{OH}: S1 Open, from V_{OL}: S1 Closed		15	35		15	30	ns
t_{RESET}	Power-Up to Registered Output High			600	1000		600	1000	ns

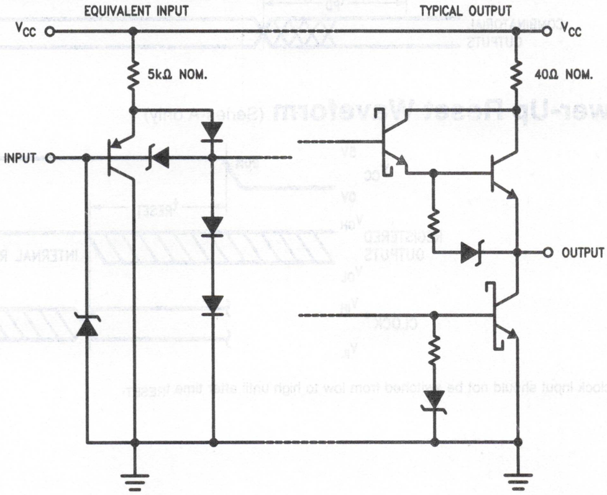
Test Load



TL/L/9998-2

MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

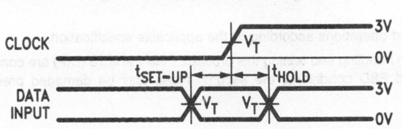
Schematic of Inputs and Outputs



TL/L/9998-3

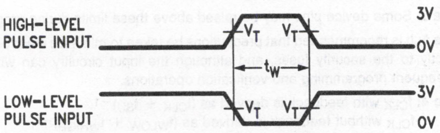
Test Waveforms

Set-Up and Hold


CLOCK
3V
0V
V_T
t_{SET-UP}
t_{HOLD}
DATA INPUT
V_T
V_T
3V
0V

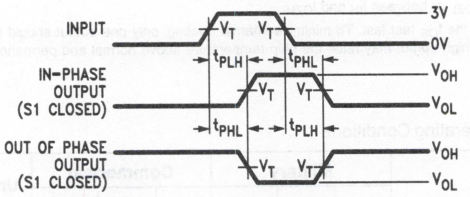
TL/L/9998-4

Pulse Width


HIGH-LEVEL PULSE INPUT
V_T
V_T
3V
0V
t_W
LOW-LEVEL PULSE INPUT
V_T
V_T
3V
0V

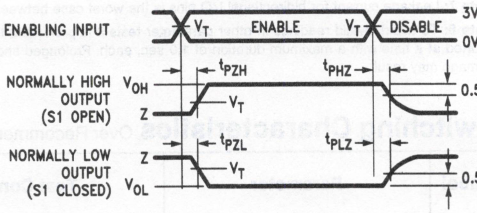
TL/L/9998-5

Propagation Delay


INPUT
V_T
V_T
3V
0V
t_{PLH}
t_{PHL}
IN-PHASE OUTPUT (S1 CLOSED)
V_T
V_T
V_{OH}
V_{OL}
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT (S1 CLOSED)
V_T
V_T
V_{OH}
V_{OL}

TL/L/9998-6

Enable and Disable


ENABLING INPUT
V_T
ENABLE
V_T
DISABLE
3V
0V
t_{PZH}
t_{PHZ}
NORMALLY HIGH OUTPUT (S1 OPEN)
V_{OH}
Z
V_T
0.5V
t_{PZL}
t_{PLZ}
NORMALLY LOW OUTPUT (S1 CLOSED)
Z
V_{OL}
V_T
0.5V

TL/L/9998-7

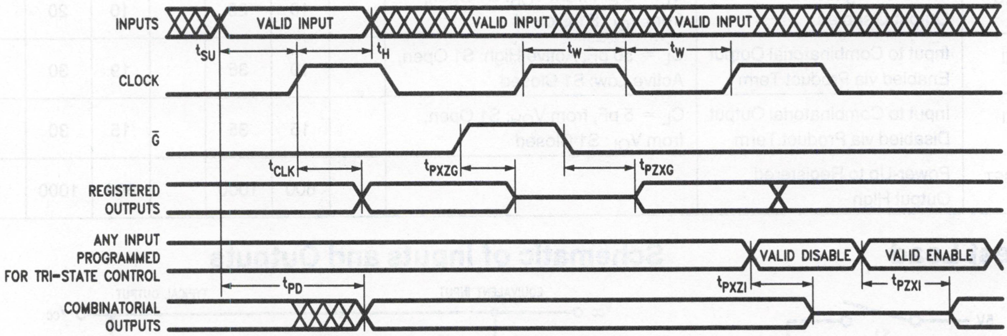
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

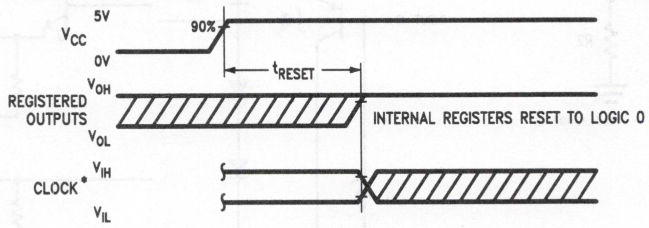
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/9998-8

Power-Up Reset Waveform (Series-A only)


5V
V_{CC}
0V
90%
t_{RESET}
V_{OH}
REGISTERED OUTPUTS
V_{OL}
INTERNAL REGISTERS RESET TO LOGIC 0
V_{IH}
CLOCK*
V_{IL}

TL/L/9998-9

*The clock input should not be switched from low to high until after time t_{RESET}.

Functional Description

All of the 24-pin Exclusive-OR PAL logic arrays consist of 20 complementary input lines and 40 product-term lines with a programmable fuse link at each intersection (1600 fuses). The product terms are organized into ten groups of four each. Three or four of the product terms in each group connect into OR-gates to produce the output logic function, depending on whether the output is combinatorial or registered.

An unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses) are in the high logic state. Therefore, if both the true and complement of at least one array input is left connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed device). Conversely, if all fuses on a product term were programmed, the product term would be held in the high state.

The exclusive-OR PAL family consists of four device types with differing mixtures of combinatorial and registered outputs. The 20L10, 20X4, 20X8 and 20X10 architectures have 0, 4, 8 and 10 registered outputs, respectively, with the balance of the 10 outputs combinatorial. All outputs are active-low and have TRI-STATE capability.

Each combinatorial output has a three product-term logic function, with the fourth product term being used for TRI-STATE control. A combinatorial output is enabled while the TRI-STATE product term is satisfied (true). Combinatorial outputs also have feedback paths from the device pins into the logic array (except for two outputs on the 20L10). This allows a pin to perform bidirectional I/O or, if the associated logic function were left unprogrammed, the output driver would remain disabled and the pin could be used as an additional dedicated input.

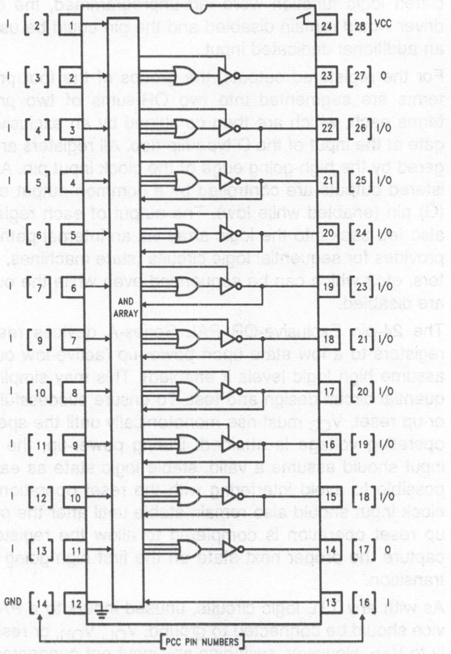
For the registered outputs, the groups of the four product terms are segmented into two OR-sums of two product terms each; which are then combined by an exclusive-OR gate at the input of the D-type flip-flop. All registers are triggered by the high-going edge of the clock input pin. All registered outputs are controlled by a common output enable ($\overline{G}$) pin (enabled while low). The output of each register is also fed back into the logic array via an internal path. This provides for sequential logic circuits (state machines, counters, etc.) which can be sequenced even while the outputs are disabled.

The 24-pin Exclusive-OR PAL Series-A devices reset all registers to a low state upon power-up (active-low outputs assume high logic levels if enabled). This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

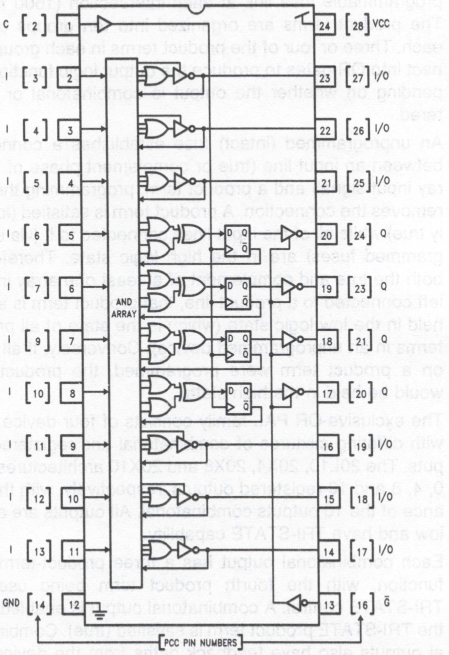
24-Pin Exclusive-OR PAL Family Block Diagrams—DIP Connections

PAL20L10


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] O
22 [26] I/O
21 [25] I/O
20 [24] I/O
19 [23] I/O
18 [21] I/O
17 [20] I/O
16 [19] I/O
15 [18] I/O
14 [17] O
13 [16] I
AND ARRAY
PCC PIN NUMBERS

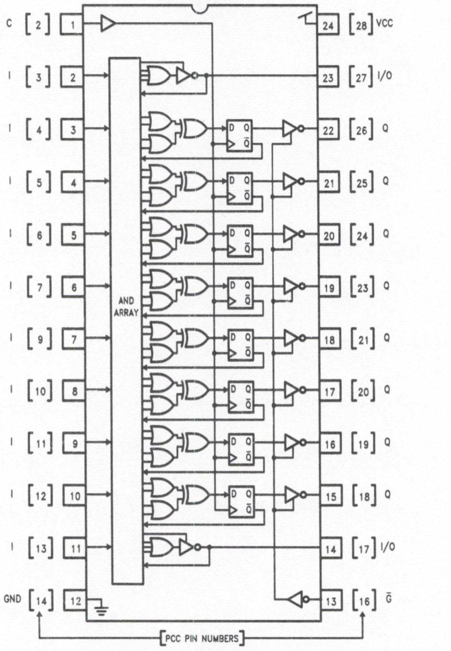
TL/L/9998-10

PAL20X4


C [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] I/O
22 [26] I/O
21 [25] I/O
20 [24] Q
19 [23] Q
18 [21] Q
17 [20] Q
16 [19] I/O
15 [18] I/O
14 [17] I/O
13 [16] G̅
D Q
Q̅
AND ARRAY
PCC PIN NUMBERS

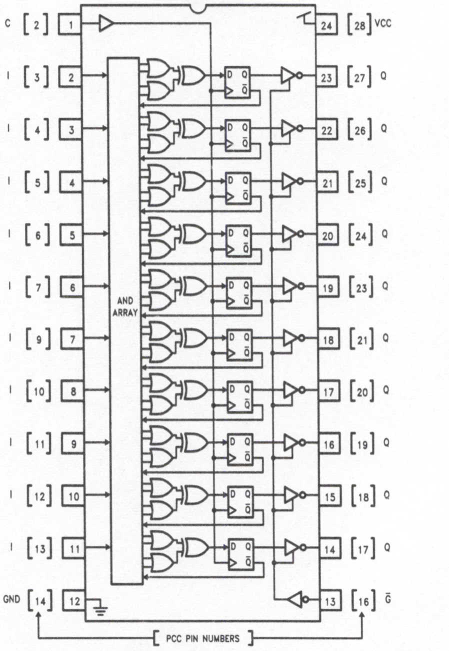
TL/L/9998-11

PAL20X8


C [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] I/O
22 [26] Q
21 [25] Q
20 [24] Q
19 [23] Q
18 [21] Q
17 [20] Q
16 [19] Q
15 [18] Q
14 [17] I/O
13 [16] G̅
D Q
Q̅
AND ARRAY
PCC PIN NUMBERS

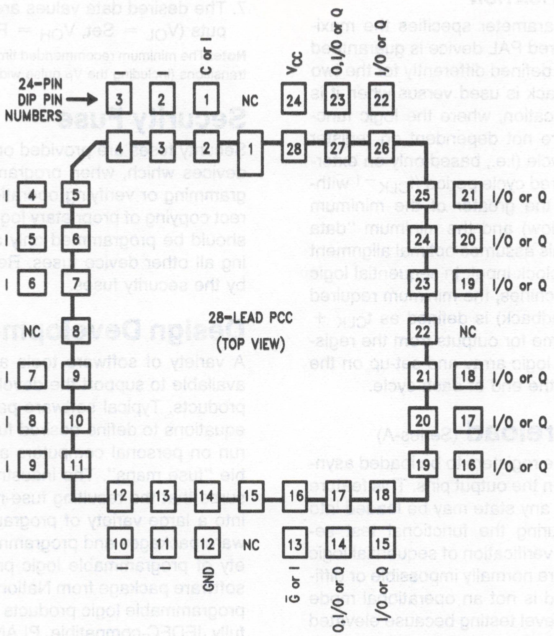
TL/L/9998-12

PAL20X10


C [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
24 [28] VCC
23 [27] Q
22 [26] Q
21 [25] Q
20 [24] Q
19 [23] Q
18 [21] Q
17 [20] Q
16 [19] Q
15 [18] Q
14 [17] Q
13 [16] G̅
D Q
Q̅
AND ARRAY
PCC PIN NUMBERS

TL/L/9998-13

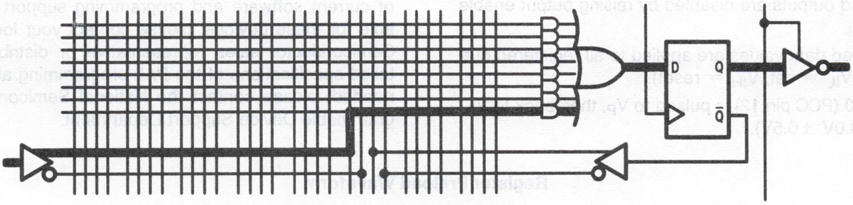
28-Lead PCC Connection Conversion Diagram*


24-PIN DIP PIN NUMBERS
C or I
V_CC
O, I/O or Q
I/O or Q
NC
3
2
1
24
23
22
4
3
2
1
28
27
26
I
4
5
25
21
I/O or Q
I
5
6
24
20
I/O or Q
I
6
7
23
19
I/O or Q
NC
8
28-LEAD PCC (TOP VIEW)
22
NC
I
7
9
21
18
I/O or Q
I
8
10
20
17
I/O or Q
I
9
11
19
16
I/O or Q
12
13
14
15
16
17
18
10
11
12
NC
13
14
15
GND
Ḡ or I
O, I/O or Q
I/O or Q

TL/L/9998-14

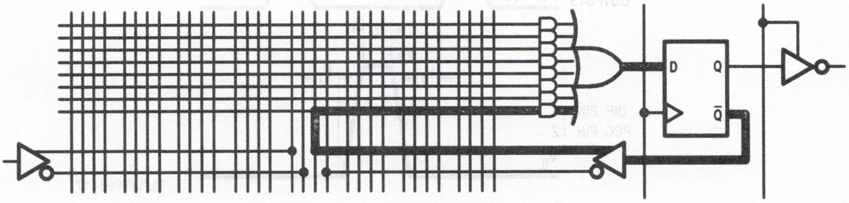
*For availability of old (Non-JEDEC) pinout, please contact your local National Semiconductor sales representative or distributor. PAL20L10 (Standard Series) is not available in the JEDEC pinout shown above.

Typical Registered Logic Function Without Feedback



TL/L/9998-15

Typical Registered Logic Function With Feedback



TL/L/9998-16

Functional Description (Continued)

CLOCK FREQUENCY SPECIFICATION

The clock frequency (f_{CLK}) parameter specifies the maximum speed at which a registered PAL device is guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, where the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e., based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_W high + t_W low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period (f_{CLK}^{-1} with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set-up on the inputs to the registers before the end of each cycle.

Output Register Preload (Series-A)

The preload function allows the registers to be loaded asynchronously from data placed on the output pins. This feature simplifies device testing since any state may be loaded into the registers at any time during the functional test sequence. This allows complete verification of sequential logic circuits, including states that are normally impossible or difficult to reach. Register preload is not an operational mode and is not intended for board level testing because elevated voltage levels are required. The programming system normally provides the preload capability as part of its functional test facility.

The register preload procedure is as follows:

1. V_{CC} is raised to 4.5V.
2. Registered outputs are disabled by raising output enable ($\overline{G}$) to V_{IH}.
3. The desired data values are applied to all registered output pins (V_{IL} = set, V_{IH} = reset).
4. DIP pin 10 (PCC pin 12) is pulsed to V_P, then back to V_{IL}. (V_P = 18.0V ±0.5V).
5. Data inputs are removed from registered output pins.
6. $\overline{G}$ is lowered to V_{IL} to enable registered outputs.
7. The desired data values are verified at all registered outputs (V_{OL} = Set, V_{OH} = Reset).

Note: The minimum recommended time delay (t_D) between successive input transitions (including the V_P pulse width on DIP pin 8) is 100 ns.

Security Fuse

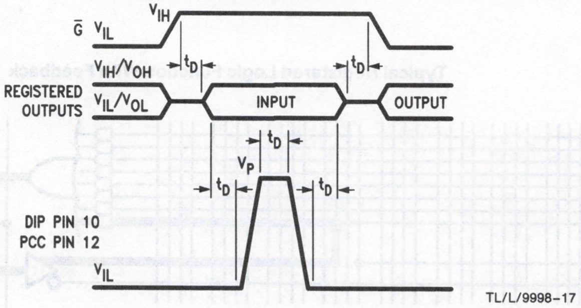
Security fuses are provided on all 24-pin Exclusive-OR PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses. Register preload is not affected by the security fuses.

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a large variety of programming equipment. Many software packages and programming units support a large variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

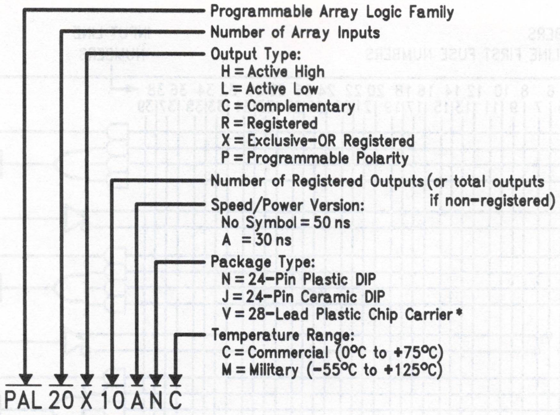
Detailed logic diagrams showing all JEDEC fuse-map addresses for the 24-pin Exclusive-OR PAL family are provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

Register Preload Waveform



Note: V_P = 18.0V ±0.5V, t_D min. = 100 ns

Ordering Information



Programmable Array Logic Family

Number of Array Inputs

Output Type:
- H = Active High
- L = Active Low
- C = Complementary
- R = Registered
- X = Exclusive-OR Registered
- P = Programmable Polarity

Number of Registered Outputs (or total outputs if non-registered)

Speed/Power Version:
- No Symbol = 50 ns
- A = 30 ns

Package Type:
- N = 24-Pin Plastic DIP
- J = 24-Pin Ceramic DIP
- V = 28-Lead Plastic Chip Carrier*

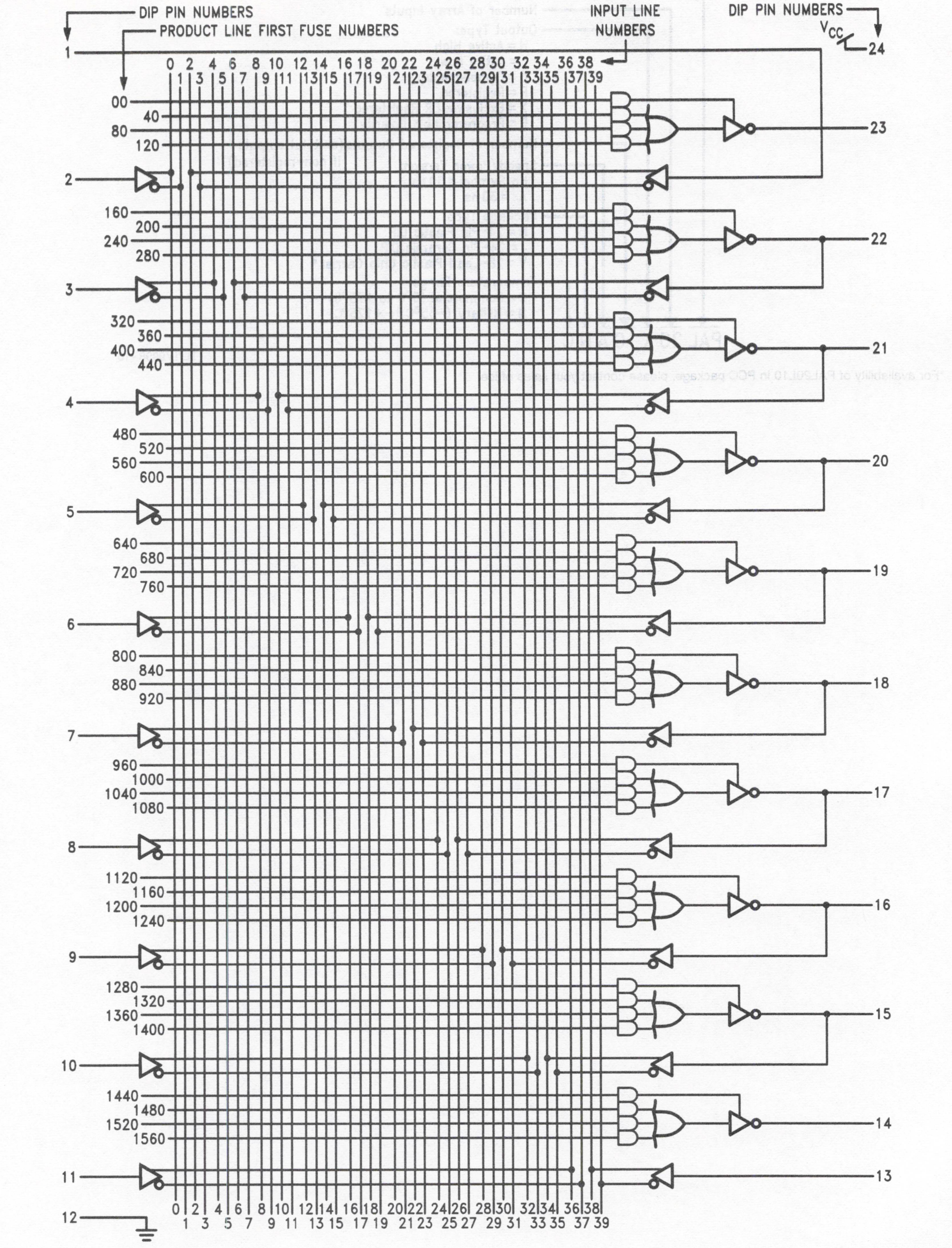
Temperature Range:
- C = Commercial (0°C to +75°C)
- M = Military (−55°C to +125°C)

PAL 20 X 10 A N C

TL/L/9998-18

*For availability of PAL20L10 in PCC package, please contact your sales office.

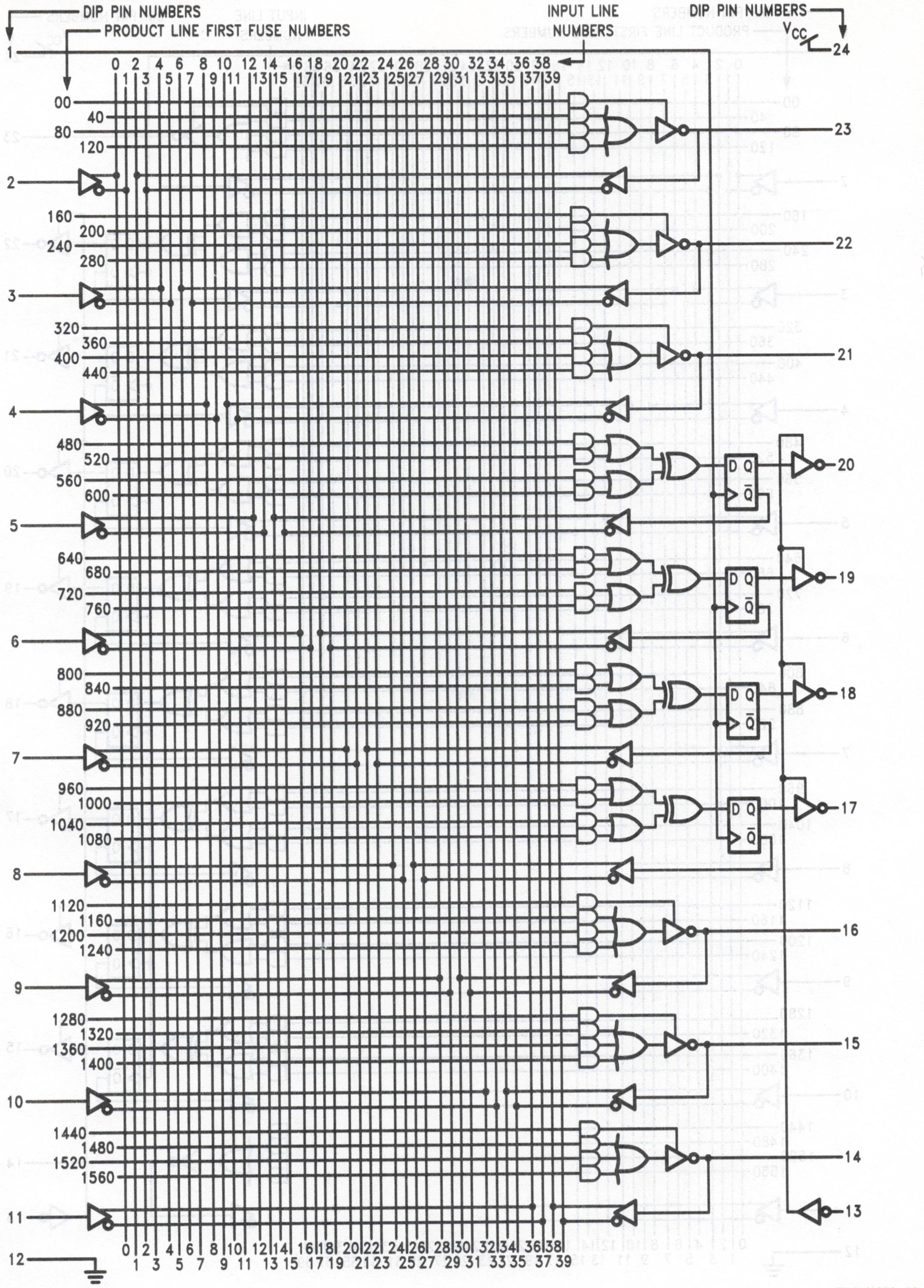
Logic Diagram PAL20L10


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
24
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
1
2
3
4
5
6
7
8
9
10
11
12
00 40 80 120
160 200 240 280
320 360 400 440
480 520 560 600
640 680 720 760
800 840 880 920
960 1000 1040 1080
1120 1160 1200 1240
1280 1320 1360 1400
1440 1480 1520 1560
23
22
21
20
19
18
17
16
15
14
13

TL/L/9998-19

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

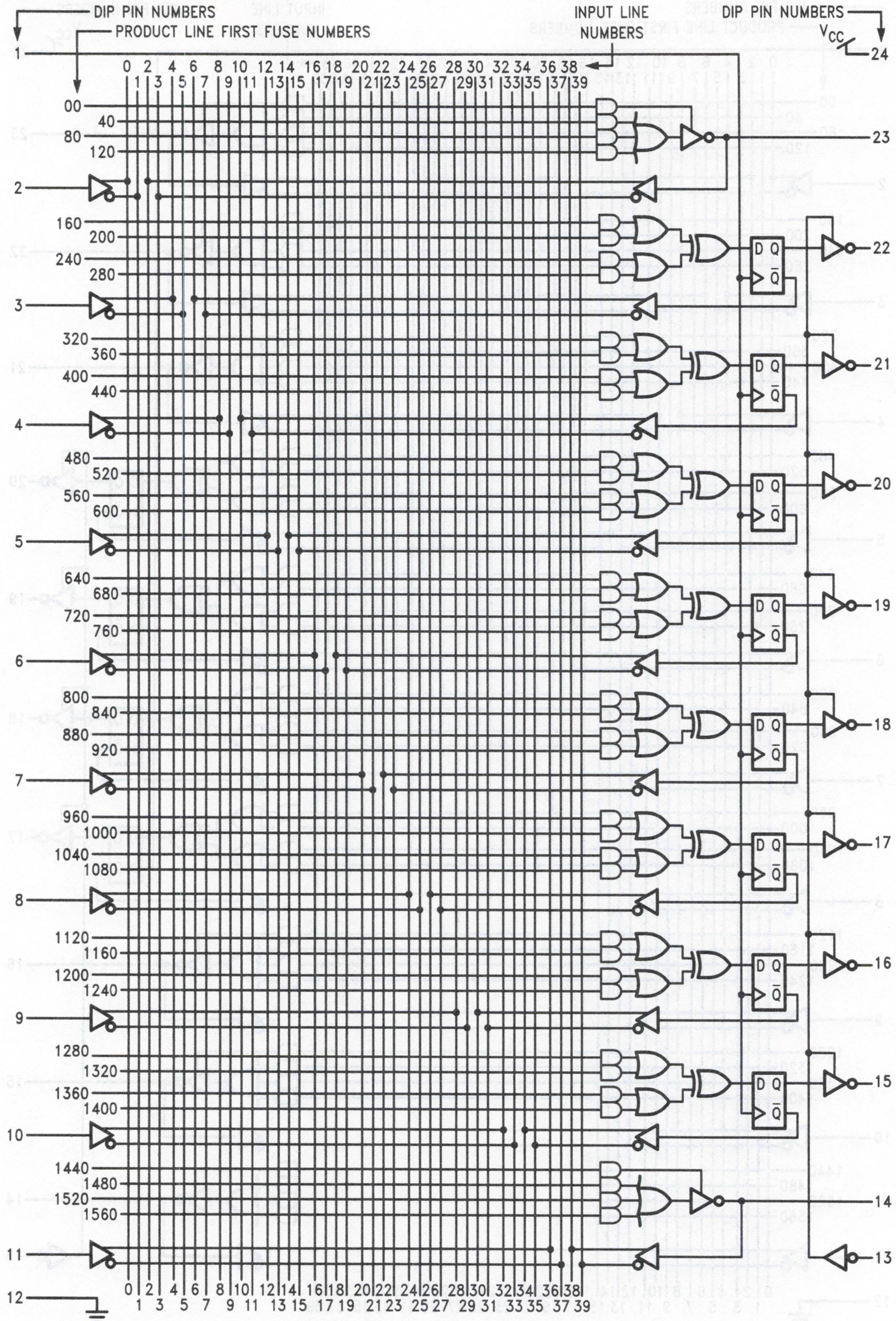
Logic Diagram PAL20X4


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
24
00
40
80
120
23
2
160
200
240
280
22
3
320
360
400
440
21
4
480
520
560
600
D Q
Q̄
20
5
640
680
720
760
19
6
800
840
880
920
18
7
960
1000
1040
1080
17
8
1120
1160
1200
1240
16
9
1280
1320
1360
1400
15
10
1440
1480
1520
1560
14
11
13
12

TL/L/9998-20

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

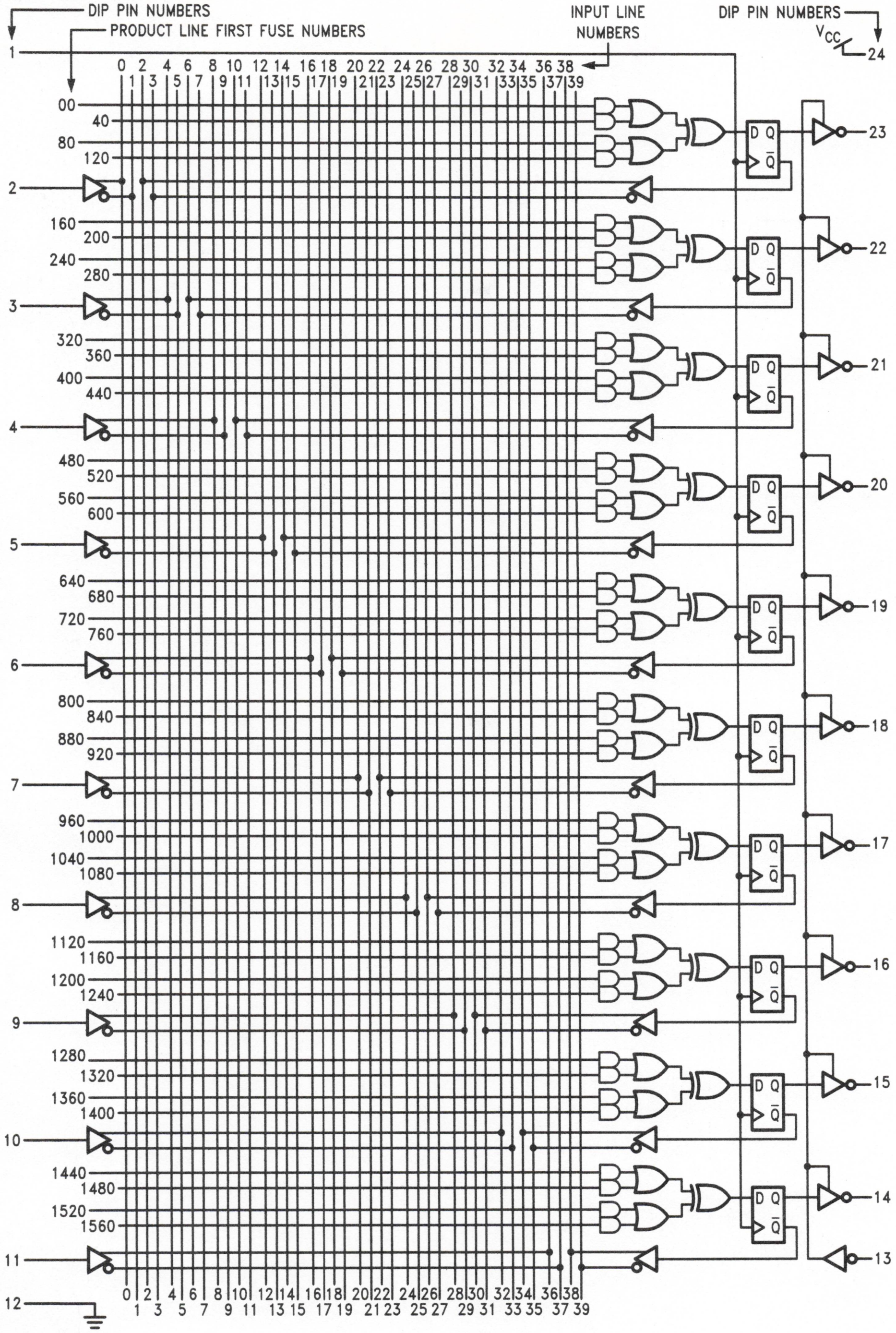
Logic Diagram PAL20X8


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
24
1
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
00
40
80
120
23
2
160
200
240
280
22
3
320
360
400
440
21
4
480
520
560
600
20
5
640
680
720
760
19
6
800
840
880
920
18
7
960
1000
1040
1080
17
8
1120
1160
1200
1240
16
9
1280
1320
1360
1400
15
10
1440
1480
1520
1560
14
11
13
12
D Q
Q̄

TL/L/9998-21

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

Logic Diagram PAL20X10


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
24
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1
2
3
4
5
6
7
8
9
10
11
12
D Q
$\overline{Q}$
23
22
21
20
19
18
17
16
15
14
13

TL/L/9998-22

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

National Semiconductor

Programmable Array Logic (PAL®) 24-Pin Medium PAL Family

General Description

The 24-pin Medium PAL family contains four of the most popular PAL architectures with speeds as fast as 10 ns maximum propagation delay. National Semiconductor's advanced Schottky TTL process with titanium tungsten fusible links is used in manufacturing the Series A and Series B devices. Series D devices are manufactured using National Semiconductor's isoplanar "FAST-Z" TTL process with highly reliable "vertical-fuse" programmable cells. Vertical fuses are implemented using avalanche-induced migration ("AIM") technology offering very high programming yields and is an extension of National's FAST® logic family. The 24-pin Medium PAL family provides high-speed user-programmable replacements for conventional SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming the programmable cells to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

The PAL logic array has a total of 20 complementary input pairs and 8 outputs generated by a single programmable AND-gate array with fixed OR-gate connections. Device outputs are either taken directly from the AND-OR functions (combinatorial) or passed through D-type flip-flops (registered). Registers allow the PAL device to implement sequential logic circuits. TRI-STATE® outputs facilitate busing and provide bidirectional I/O capability. The medium PAL family offers a variety of combinatorial and registered output mixtures, as shown in the Device Types table below.

On power-up, Series-D devices reset all registers to simplify sequential circuit design and testing and for Series B devices, the registers are set on power-up. For Series D and Series B devices, direct register preload is also provided to facilitate device testing. Security fuses can be programmed to prevent direct copying of proprietary logic patterns.

Features

- As fast as 10 ns maximum propagation delay (combinatorial)
- User-programmable replacement for TTL logic
- High programming yield and reliability of vertical-fuse technology for Series D products. (Programming equipment with certified vertical-fuse algorithm required)
- Extension of FAST product line (Series-D).
- Large variety of JEDEC-compatible programming equipment and design development software available
- Fully supported by National PLAN™ software
- Power-up set/reset for registered outputs (Series-B, D)
- Register preload facilitates device testing (Series-B, D)
- Security fuse prevents direct copying of logic patterns

Device Types

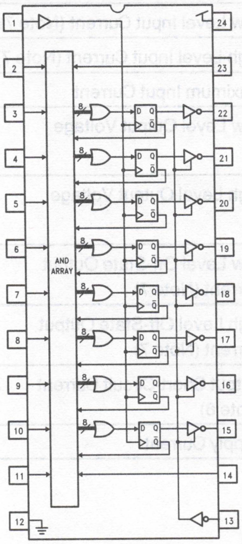
Device Type	Dedicated Inputs	Registered Outputs (With Feedback)	Combinatorial	
			I/Os	Outputs
PAL20L8	14	—	6	2
PAL20R4	12	4	4	—
PAL20R6	12	6	2	—
PAL20R8	12	8	—	—

Speed/Power Versions

Series	Example	Commercial		Military	
		t_{PD}	I_{CC}	t_{PD}	I_{CC}
A	PAL 20L8A	25 ns	210 mA	30 ns	210 mA
B	PAL 20L8B	15 ns	210 mA	20 ns	210 mA
D*	PAL20L8D	10 ns	210 mA		

*Preliminary

Block Diagram—PAL20R8



TL/L/9394-1

Series A (PAL20L8A, PAL20R4A, PAL20R6A, PAL20R8A)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +5.5V
Off-State Output Voltage(Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance (Note 3)	400V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	20	7		15	7		ns
		High	20	7		15	7		ns
t_{SU}	Setup Time from Input or Feedback to Clock		30	18		25	18		ns
t_H	Hold Time of Input after Clock		0	−10		0	−10		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		33.3	20		33.3	28.5	MHz
		Without Feedback		71.4	25		71.4	33.3	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	3.4		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	µA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	µA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				160	210	mA

Series A (PAL20L8A, PAL20R4A, PAL20R6A, PAL20R8A) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 (DIP) are connected directly to the security fuses, and although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

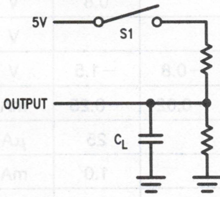
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commerical			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		18	30		18	25	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		12	20		12	15	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	25		10	20	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		11	25		11	20	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	30		10	25	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		13	30		13	25	ns

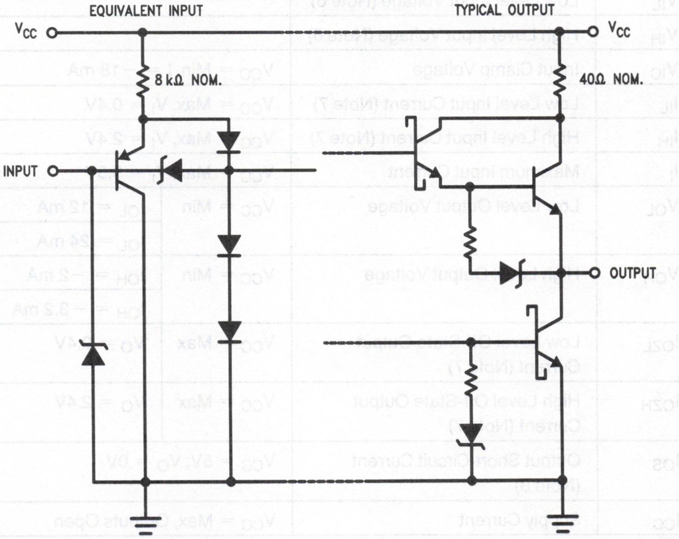
Test Load



TL/L/9394-2

MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

Schematic of Inputs and Outputs



TL/L/9394-28

Series B (PAL20L8B, PAL20R4B, PAL20R6B, PAL20R8B)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +5.5V
Off-State Output Voltage(Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance (Note 3)	2000V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	12	8		10	5		ns
		High	12	8		10	5		ns
t_{SU}	Setup Time from Input or Feedback to Clock		20	10		15	10		ns
t_H	Hold Time of Input after Clock		0	−5		0	−5		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		55.5	28.5		55.5	37	MHz
		Without Feedback		62.5	41.7		100	50	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)						0.8	V
V_{IH}	High Level Input Voltage (Note 6)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V				−0.02	−0.25	mA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V					25	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	3.4		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 0.4V				−100	µA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max	V_O = 2.4V				100	µA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				160	210	mA

Series B (PAL20L8B, PAL20R4B, PAL20R6B, PAL20R8B) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 (DIP) are connected directly to the security fuses, and although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged preventing subsequent programming and verification operations.

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

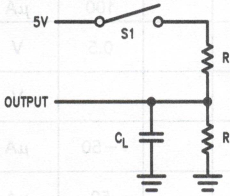
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commerical			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed		11	20		11	15	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed		8	15		8	12	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	20		10	15	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		8	20		8	12	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		11	25		11	18	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		11	20		11	15	ns
t_{SET}	Power-Up to Registered Output Low			600	1000		600	1000	ns

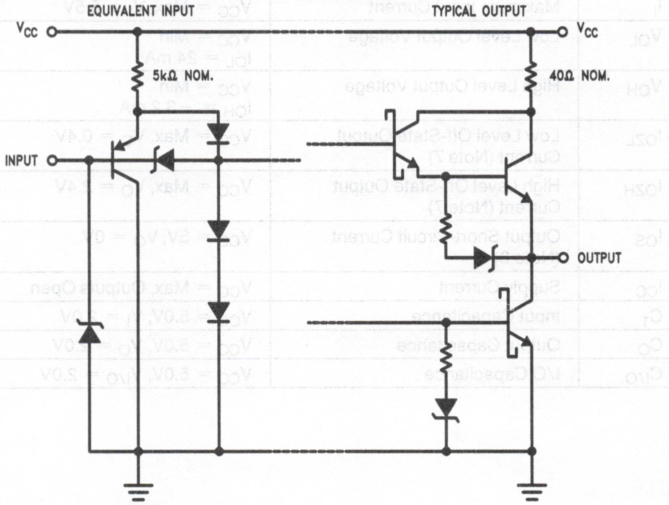
Test Load



TL/L/9394-2

MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

Schematic of Inputs and Outputs



TL/L/9394-7

Series D (PAL20L8D, PAL20R4D, PAL20R6D, PAL20R8D) **PRELIMINARY**

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +7.0V
Off-State Output Voltage (V_O) (Notes 2 & 3)	−1.5V to +5.5V
Input Current (Note 2)	−30 mA to +5.0 mA
Output Current (I_{OL})	+100 mA

Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance	TBD

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Commercial Min	Commercial Nom	Commercial Max	Units
V_{CC}	Supply Voltage		4.75	5	5.25	V
T_A	Operating Free-Air Temperature		0	25	75	°C
t_W	Clock Pulse Width	Low	7			ns
		High	7			ns
t_{SU}	Setup Time from Input or Feedback to Clock		10			ns
t_H	Hold Time of Input after Clock		0			ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback			55.5	MHz
		Without Feedback			71.4	MHz
V_Z	Register Preload Control Voltage		9.5	9.75	10.0	V

Electrical Characteristics Over Recommended Operating Conditions (Note 5)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 6)				0.8	V
V_{IH}	High Level Input Voltage (Note 6)		2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA			−1.2	V
I_{IL}	Low Level Input Current (Note 7)	V_{CC} = Max, V_I = 0.4V			−250	μA
I_{IH}	High Level Input Current (Note 7)	V_{CC} = Max, V_I = 2.4V			25	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V			100	μA
V_{OL}	Low Level Output Voltage	V_{CC} = Min I_{OL} = 24 mA			0.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min I_{OH} = −3.2 mA	2.7			V
I_{OZL}	Low Level Off-State Output Current (Note 7)	V_{CC} = Max, V_O = 0.4V			−50	μA
I_{OZH}	High Level Off-State Output Current (Note 7)	V_{CC} = Max, V_O = 2.4V			50	μA
I_{OS}	Output Short-Circuit Current (Note 8)	V_{CC} = 5V, V_O = 0V	−50		−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open		125	210	mA
C_1	Input Capacitance	V_{CC} = 5.0V, V_I = 2.0V		8		pF
C_O	Output Capacitance	V_{CC} = 5.0V, V_O = 2.0V		8		pF
$C_{I/O}$	I/O Capacitance	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V		8		pF

Series D (PAL20L8D, PAL20R4D, PAL20R6D, PAL20R8D) (Continued)

PRELIMINARY

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: V_O must not exceed $V_{CC} + 1V$

Note 4: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 5: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 6: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

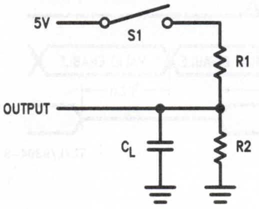
Note 7: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 8: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Commerical			Units
			Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	$C_L = 50$ pF, S1 Closed			10	ns
t_{CLK}	Clock to Registered Output or Feedback	$C_L = 50$ pF, S1 Closed			8	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed			10	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed			10	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed			10	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed			10	ns
t_{RESET}	Power-Up to Registered Output High				1000	ns

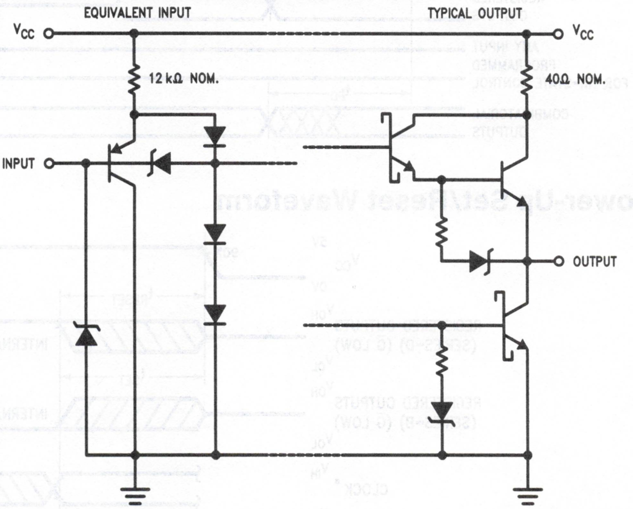
Test Load



TL/L/9394-23

R1 = 200
R2 = 390

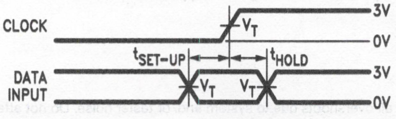
Schematic of Inputs and Outputs



TL/L/9394-24

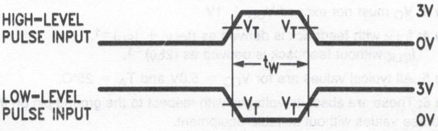
Test Waveforms

Set-Up and Hold


CLOCK
DATA INPUT
3V
0V
3V
0V
V_T
t_{SET-UP}
t_{HOLD}
V_T
V_T

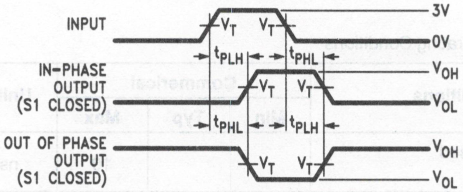
TL/L/9394-3

Pulse Width


HIGH-LEVEL PULSE INPUT
LOW-LEVEL PULSE INPUT
3V
0V
3V
0V
V_T
V_T
t_W
V_T
V_T

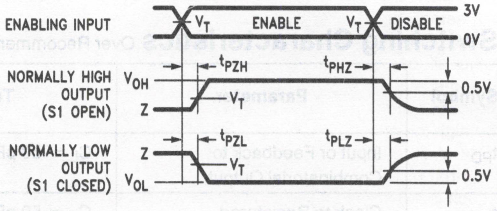
TL/L/9394-5

Propagation Delay


INPUT
IN-PHASE OUTPUT (S1 CLOSED)
OUT OF PHASE OUTPUT (S1 CLOSED)
3V
0V
V_{OH}
V_{OL}
V_{OH}
V_{OL}
V_T
V_T
t_{PLH}
t_{PHL}
t_{PHL}
t_{PLH}

TL/L/9394-4

Enable and Disable


ENABLING INPUT
NORMALLY HIGH OUTPUT (S1 OPEN)
NORMALLY LOW OUTPUT (S1 CLOSED)
ENABLE
DISABLE
3V
0V
V_T
V_T
V_{OH}
Z
Z
V_{OL}
t_{PZH}
t_{PHZ}
t_{PZL}
t_{PLZ}
0.5V
0.5V

TL/L/9394-6

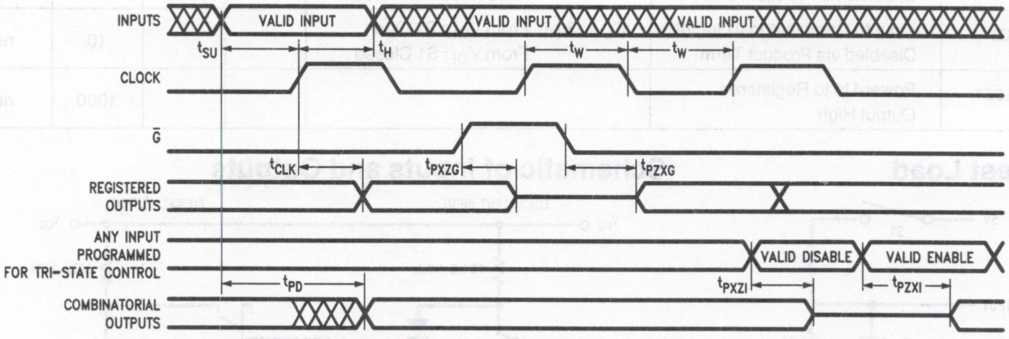
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

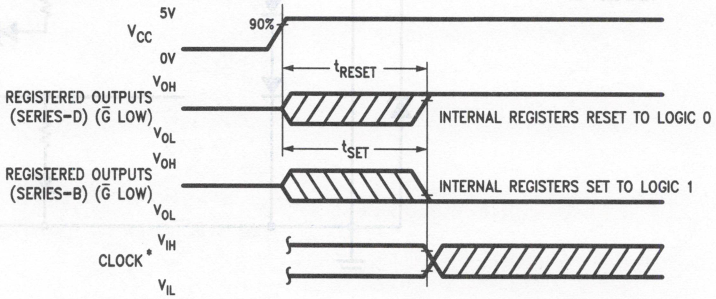
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/9394-8

Power-Up Set/Reset Waveform


5V
V_{CC}
0V
90%
t_{RESET}
REGISTERED OUTPUTS (SERIES-D) ($\overline{G}$ LOW)
V_{OH}
V_{OL}
INTERNAL REGISTERS RESET TO LOGIC 0
t_{SET}
REGISTERED OUTPUTS (SERIES-B) ($\overline{G}$ LOW)
V_{OH}
V_{OL}
INTERNAL REGISTERS SET TO LOGIC 1
CLOCK*
V_{IH}
V_{IL}

TL/L/9394-9

*The clock input should not be switched from low to high until after time t_{RESET} or t_{SET}.

Functional Description

All of the 24-pin Medium PAL logic arrays consist of 20 complementary input lines and 64 product-term lines with a programmable cell at each intersection (2560 cells). The product terms are organized into eight groups of eight each. Seven or eight of the product terms in each group connect into an OR-gate to produce the sum-of-products logic function, depending on whether the output is combinatorial or registered.

For the fuse-link PAL devices (all PAL devices prior to National Series D), an unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. In the National Series D vertical fuse PAL devices, a programmed vertical fuse cell establishes a connection between an input line and a product term. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses for the fuse-link devices, or by programming the corresponding cells for the vertical fuse devices) are in the high logic state. Therefore, if both the true and complement of at least one array input is connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed fuse-link device). Conversely, if all input lines are disconnected from a product line, the product term and the resulting logic function would be held in the high state (which is the state of all product terms in an unprogrammed National Series-D PAL device).

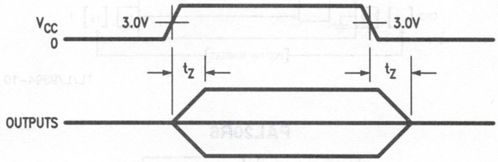
The medium PAL family consists of four device types with differing mixtures of combinatorial and registered outputs. The 20L8, 20R4, 20R6 and 20R8 architectures have 0, 4, 6 and 8 registered outputs, respectively, with the balance of the 8 outputs combinatorial. All outputs are active-low and have TRI-STATE capability.

Each combinatorial output has a seven product-term logic function, with the eighth product term being used for TRI-STATE control. A combinatorial output is enabled while the TRI-STATE product term is satisfied (true). Combinatorial outputs also have feedback paths from the device pins into the logic array (except for two outputs on the 20L8). This allows a pin to perform bidirectional I/O or, if the associated TRI-STATE control product term were programmed to remain unsatisfied (always false), the output driver would remain disabled and the pin could be used as an additional dedicated input.

Registered outputs each have an eight product-term logic function feeding into a D-type flip-flop. All registers are triggered by the high-going edge of the clock input pin. All registered outputs are controlled by a common output enable ($\overline{G}$) pin (enabled while low). The output of each register is also fed back into the logic array via an internal path. This provides for sequential logic circuits (state machines, counters, etc.) which can be sequenced even while the outputs are disabled.

Series-B Medium PAL devices set all the registers high on power-up (active low outputs assume low logic levels if enabled). Series-D Medium PAL devices reset all registers to a low state upon power-up (active-low outputs assume high logic levels if enabled). This may simplify sequential circuit design and test. To ensure successful power-up set or reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible to avoid interfering with the set or reset operation. The clock input should also remain stable until after the power-up set or reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

For the National Series D PAL device, during power-up, all outputs are held in the high-impedance state until DC power supply conditions are met (V_{CC} approximately 3.0V), after which they may be enabled by the TRI-STATE control product terms (combinatorial outputs) or the $\overline{G}$ pin (registered outputs). Whenever V_{CC} goes below 3V (at 25°C), the outputs are disabled as shown in *Figure 1* below.



TL/L/9394–25

Note: t_Z is less than 100 ns.

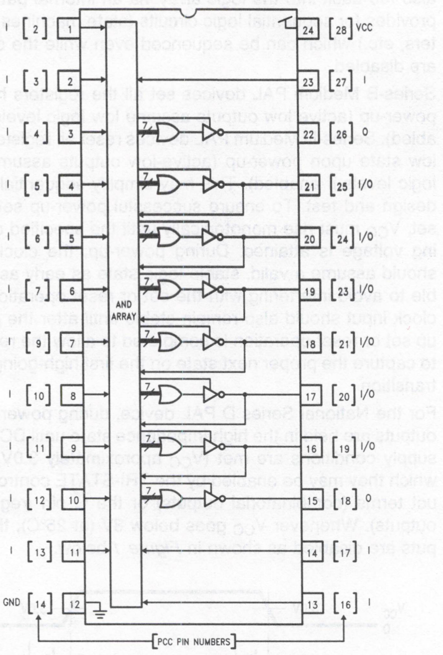
FIGURE 1. Series-D Power-Up TRI-STATE Waveform

In an unprogrammed National Series D PAL device, no array inputs are connected to any product-term lines. Therefore, all combinatorial outputs would be enabled and driving low logic levels (after power-up is completed). All registers would still initialize to the low state, but would become permanently set (low-level outputs, if enabled) following the first clock transition.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

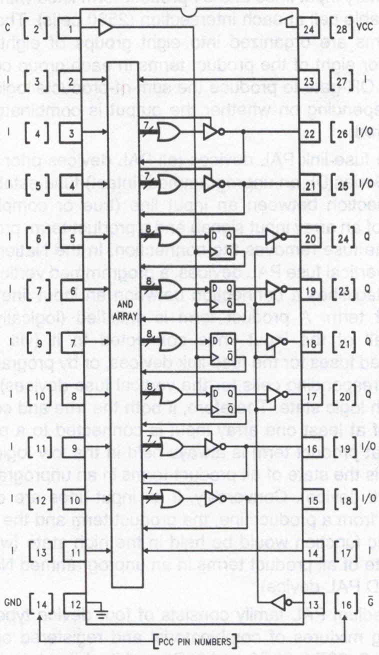
24-Pin Medium PAL Family Block Diagrams—DIP Connections

PAL20L8


I [2] 1
24 [28] VCC
I [3] 2
23 [27] I
I [4] 3
22 [26] O
I [5] 4
21 [25] I/O
I [6] 5
20 [24] I/O
I [7] 6
19 [23] I/O
I [9] 7
18 [21] I/O
I [10] 8
17 [20] I/O
I [11] 9
16 [19] I/O
I [12] 10
15 [18] O
I [13] 11
14 [17] I
GND [14] 12
13 [16] I
AND ARRAY
7
PCC PIN NUMBERS

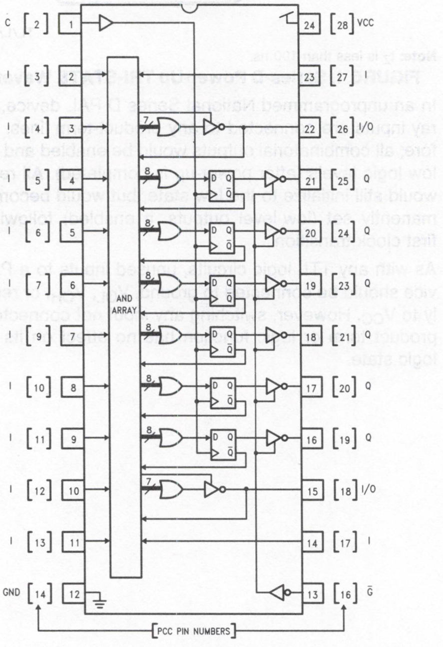
TL/L/9394-10

PAL20R4


C [2] 1
24 [28] VCC
I [3] 2
23 [27] I
I [4] 3
22 [26] I/O
I [5] 4
21 [25] I/O
I [6] 5
20 [24] Q
I [7] 6
19 [23] Q
I [9] 7
18 [21] Q
I [10] 8
17 [20] Q
I [11] 9
16 [19] I/O
I [12] 10
15 [18] I/O
I [13] 11
14 [17] I
GND [14] 12
13 [16] G̅
AND ARRAY
7
8
D Q
Q̅
PCC PIN NUMBERS

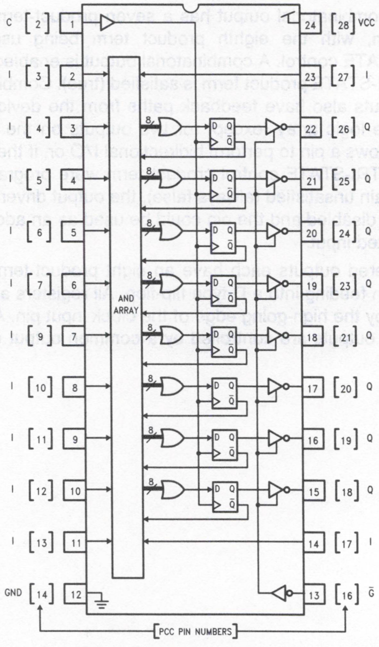
TL/L/9394-11

PAL20R6


C [2] 1
24 [28] VCC
I [3] 2
23 [27] I
I [4] 3
22 [26] I/O
I [5] 4
21 [25] Q
I [6] 5
20 [24] Q
I [7] 6
19 [23] Q
I [9] 7
18 [21] Q
I [10] 8
17 [20] Q
I [11] 9
16 [19] Q
I [12] 10
15 [18] I/O
I [13] 11
14 [17] I
GND [14] 12
13 [16] G̅
AND ARRAY
7
8
D Q
Q̅
PCC PIN NUMBERS

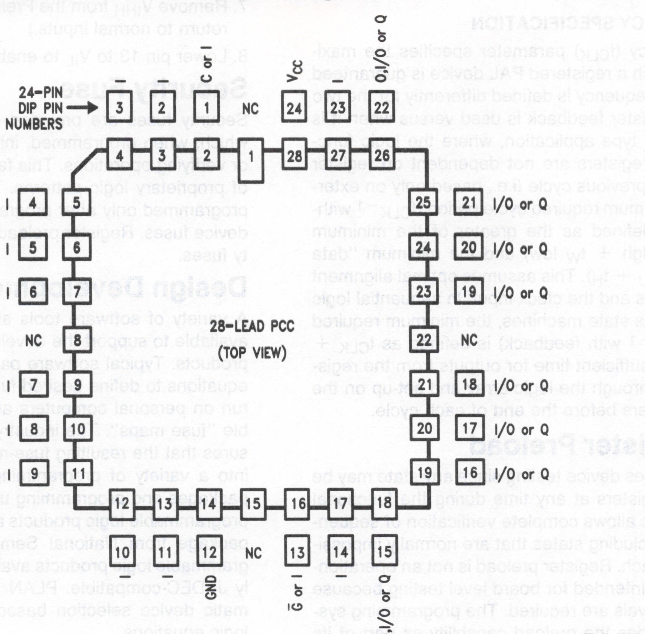
TL/L/9394-12

PAL20R8


C [2] 1
24 [28] VCC
I [3] 2
23 [27] I
I [4] 3
22 [26] Q
I [5] 4
21 [25] Q
I [6] 5
20 [24] Q
I [7] 6
19 [23] Q
I [9] 7
18 [21] Q
I [10] 8
17 [20] Q
I [11] 9
16 [19] Q
I [12] 10
15 [18] Q
I [13] 11
14 [17] I
GND [14] 12
13 [16] G̅
AND ARRAY
8
D Q
Q̅
PCC PIN NUMBERS

TL/L/9394-13

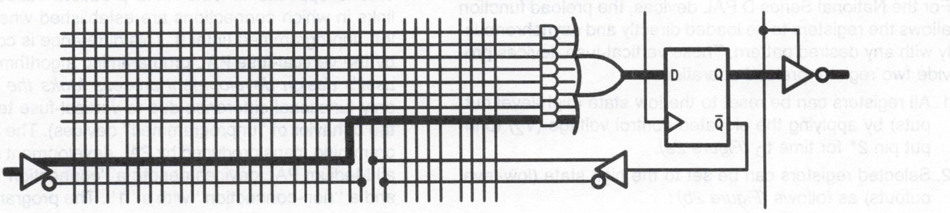
28-Lead PCC Connection Conversion Diagram*


C or I
VCC
O1/I/O or Q
24-PIN DIP PIN NUMBERS
3
2
1
NC
24
23
22
4
3
2
1
28
27
26
I 4 5
I 5 6
I 6 7
NC 8
7 9
8 10
9 11
25 21 I/O or Q
24 20 I/O or Q
23 19 I/O or Q
22 NC
21 18 I/O or Q
20 17 I/O or Q
19 16 I/O or Q
28-LEAD PCC (TOP VIEW)
12
13
14
15
16
17
18
10
11
12
NC
13
14
15
GND
G or I
O1/I/O or Q

TL/L/9394–14

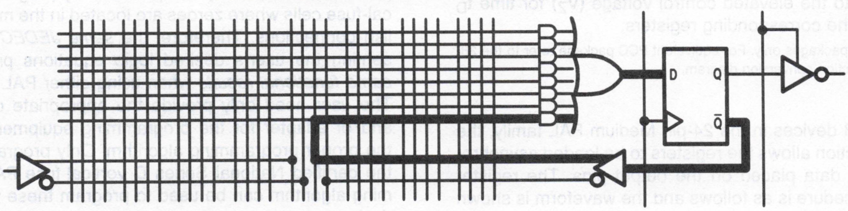
*For availability of old (non-JEDEC) pinout, please contact your local National Semiconductor sales representative or distributor.

Typical Registered Logic Function Without Feedback



TL/L/9394–15

Typical Registered Logic Function With Feedback



TL/L/9394–16

Functional Description (Continued)

CLOCK FREQUENCY SPECIFICATION

The clock frequency (f_{CLK}) parameter specifies the maximum speed at which a registered PAL device is guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, where the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e., based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_W high + t_W low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period (f_{CLK}^{-1} with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set-up on the inputs to the registers before the end of each cycle.

Output Register Preload

This feature simplifies device testing since any state may be loaded into the registers at any time during the functional test sequence. This allows complete verification of sequential logic circuits, including states that are normally impossible or difficult to reach. Register preload is not an operational mode and is not intended for board level testing because elevated voltage levels are required. The programming system normally provides the preload capability as part of its functional test facility. This feature is available on the Series-B and Series-D devices only.

SERIES-D

For the National Series D PAL devices, the preload function allows the registers to be loaded directly and asynchronously with any desired pattern. These vertical-fuse devices provide two register preload operations:

1. All registers can be reset to the low state (high-level outputs) by applying the elevated control voltage (V_Z) to input pin 2* for time t_D *(Figure 2a)*.
2. Selected registers can be set to the high state (low-level outputs) as follows *(Figure 2b)*:
   a. All registered outputs are disabled by raising the $\overline{G}$ input pin 1* to V_{IH}
   b. After time t_D, the selected registered output pins are raised to the elevated control voltage (V_Z) for time t_D to set the corresponding registers.

*Refers to DIP packages only. For equivalent PCC package refer to the 28-pin PCC connection conversion diagram.

SERIES-B

For Series-B devices in the 24-pin Medium PAL family, the preload function allows the registers to be loaded asynchronously from data placed on the output pins. The register preload procedure is as follows and the waveform is shown in *Figure 3*:

1. Apply V_{CC}
2. Disable the registered outputs by raising pin 13 to V_{IH}.
3. Apply V_{IL} to inputs corresponding to all non-registered outputs.
4. Apply the desired V_{IL}/V_{IH} to the inputs corresponding to registered outputs. (A high input will force the register high and the output low.)
5. Raise the Preload pins (pin 18 and pin 14) to V_{IHH}. ($V_{IHH} = 11.75V \pm 0.25V$)
6. Apply a clock pulse.
7. Remove V_{IHH} from the Preload pins. (The data inputs will return to normal inputs.)
8. Lower pin 13 to V_{IL} to enable the outputs.

Security Fuse

Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses. Register preload is not affected by the security fuses.

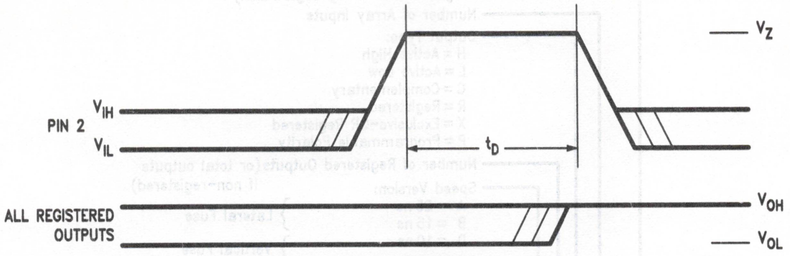
Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a variety of programming equipment. Many software packages and programming units support a wide variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

In National Series D PAL devices, logical and physical connections between array input lines and product-term lines are established when vertical fuse cells are programmed. This is opposite to other PAL products based on fusible links in which connections are established when fuses are left unprogrammed (intact). This difference is compensated by the vertical-fuse PAL programming algorithm so that the *user's design development process looks the same.* (The only functional difference due to vertical-fuse technology is the behavior of "unprogrammed" devices). The JEDEC programming maps produced by PAL development software for all Medium PAL devices denote a "connection" with a "0", and a "non-connection" with a "1". The programming algorithms for most fuse-link PLDs program fuses where ones are located in the map to remove corresponding connections, whereas the algorithm for National Series D PAL products automatically compensates by programming vertical-fuse cells where zeroes are located in the map to establish connections. Therefore, the *same JEDEC map* representing the user's desired logic equations produces the *same functional results* when using either PAL technology. The user need only provide the appropriate device code and/or adapter for the programming equipment to invoke the proper programming algorithm. Only programmers with the certified National Series D vertical-fuse PAL programming algorithm can be used to program these vertical-fuse devices.

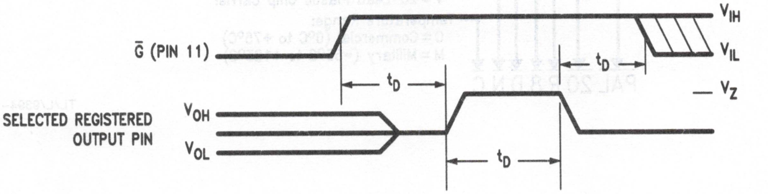
Detailed logic diagrams showing all JEDEC fuse-map addresses for the 24-pin Medium PAL family are provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

Functional Description (Continued)


V_Z
V_{IH}
PIN 2
V_{IL}
t_D
ALL REGISTERED OUTPUTS
V_{OH}
V_{OL}

TL/L/9394-26

a) To Reset All Registers

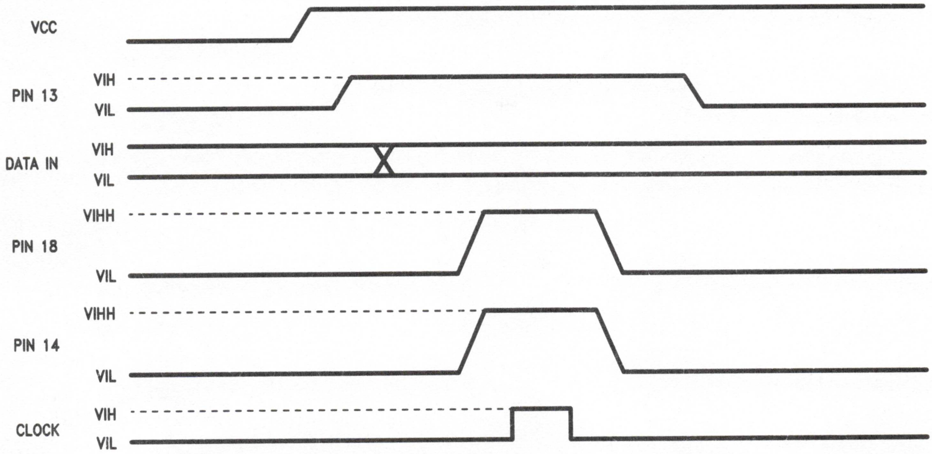

V_{IH}
$\overline{G}$ (PIN 11)
V_{IL}
t_D
t_D
V_Z
SELECTED REGISTERED OUTPUT PIN
V_{OH}
V_{OL}
t_D

TL/L/9394-27

b) To Set Selected Registers

Note: V_Z = 9.5V to 10.0V, t_D min. = 500 ns

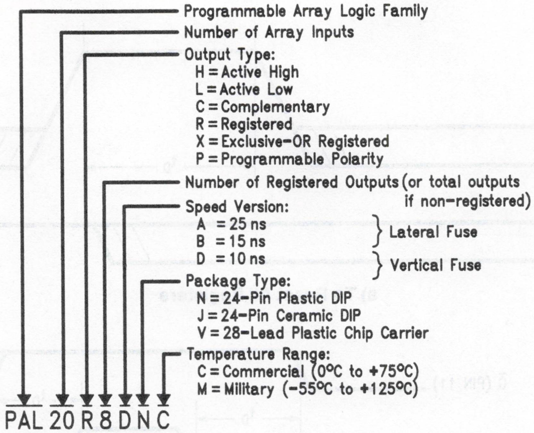
FIGURE 2. Series-D Register Preload Waveforms (Vertical-Fuse Devices)


VCC
PIN 13
VIH
VIL
DATA IN
VIH
VIL
PIN 18
VIHH
VIL
PIN 14
VIHH
VIL
CLOCK
VIH
VIL

TL/L/9394-29

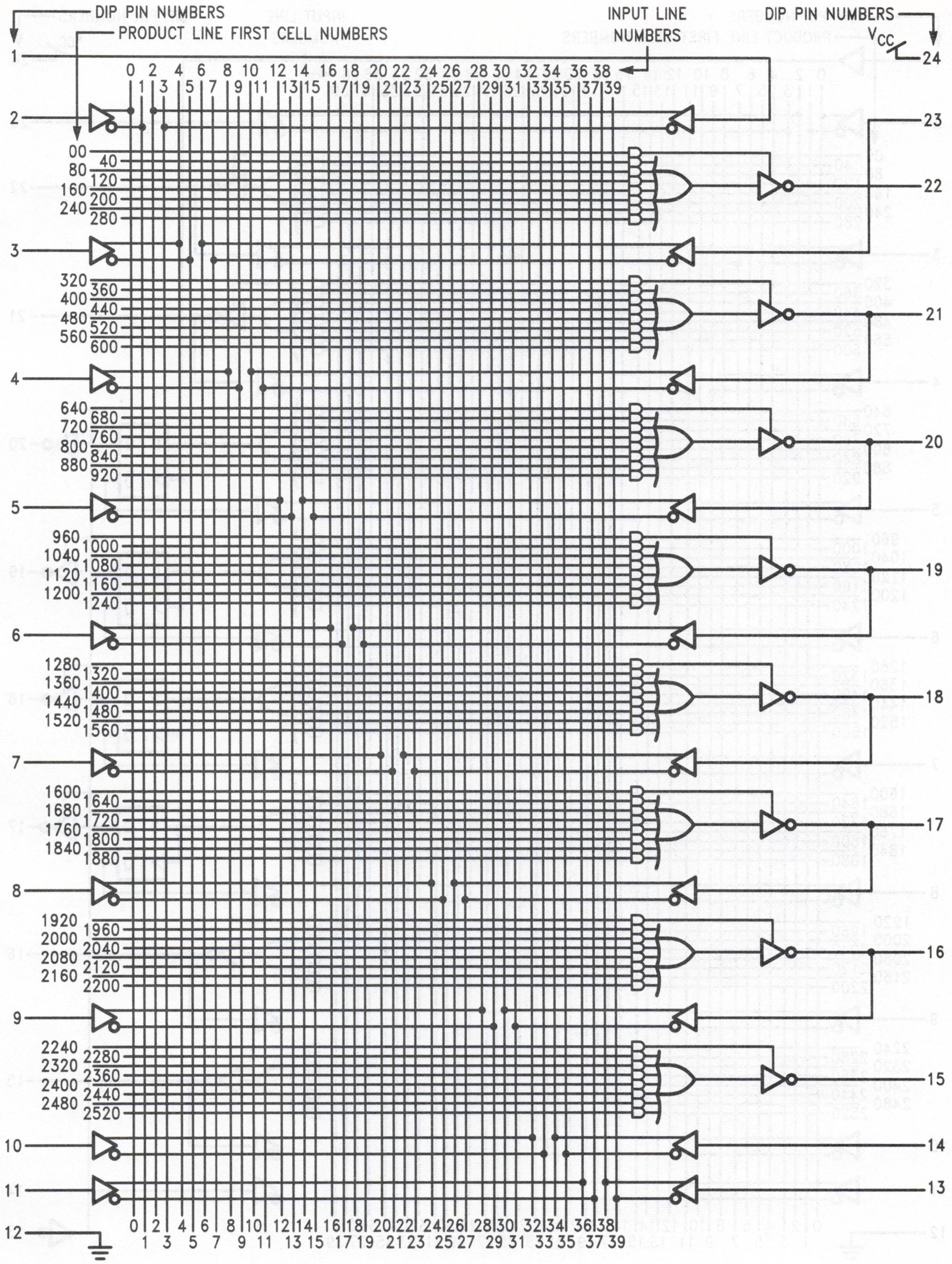
FIGURE 3. Series-B Register Preload Waveform (Fuse-Link Devices)

Ordering Information


Programmable Array Logic Family
Number of Array Inputs
Output Type:
H = Active High
L = Active Low
C = Complementary
R = Registered
X = Exclusive-OR Registered
P = Programmable Polarity
Number of Registered Outputs (or total outputs if non-registered)
Speed Version:
A = 25 ns
B = 15 ns
D = 10 ns
Lateral Fuse
Vertical Fuse
Package Type:
N = 24-Pin Plastic DIP
J = 24-Pin Ceramic DIP
V = 28-Lead Plastic Chip Carrier
Temperature Range:
C = Commercial (0°C to +75°C)
M = Military (−55°C to +125°C)
PAL 20 R 8 D N C

TL/L/9394-18

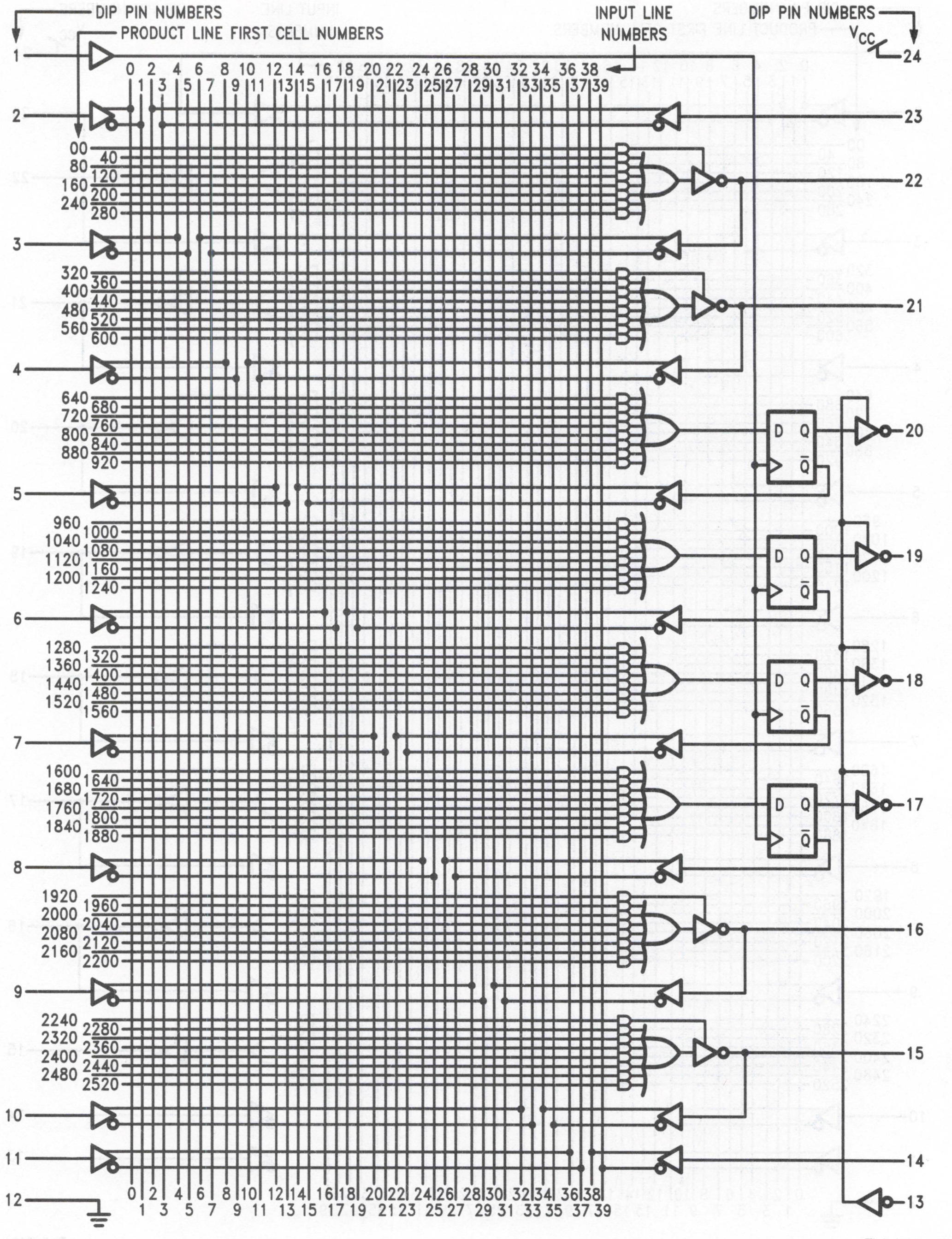
Logic Diagram—PAL20L8


DIP PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
00 40 80 120 160 200 240 280
320 360 400 440 480 520 560 600
640 680 720 760 800 840 880 920
960 1000 1040 1080 1120 1160 1200 1240
1280 1320 1360 1400 1440 1480 1520 1560
1600 1640 1680 1720 1760 1800 1840 1880
1920 1960 2000 2040 2080 2120 2160 2200
2240 2280 2320 2360 2400 2440 2480 2520

TL/L/9394–19

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

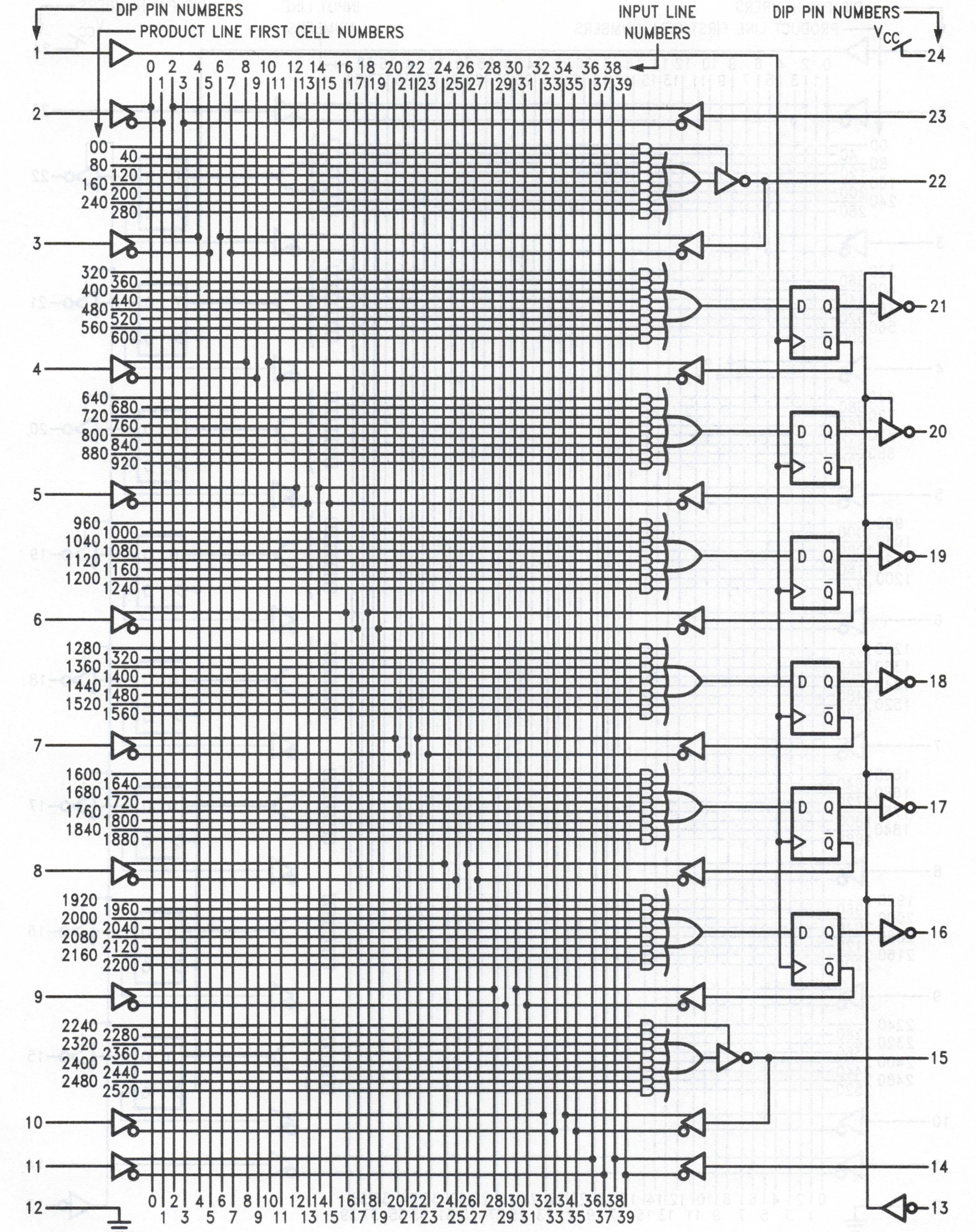
Logic Diagram—PAL20R4


DIP PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000
2040
2080
2120
2160
2200
2240
2280
2320
2360
2400
2440
2480
2520
D
Q
Q̄

TL/L/9394-20

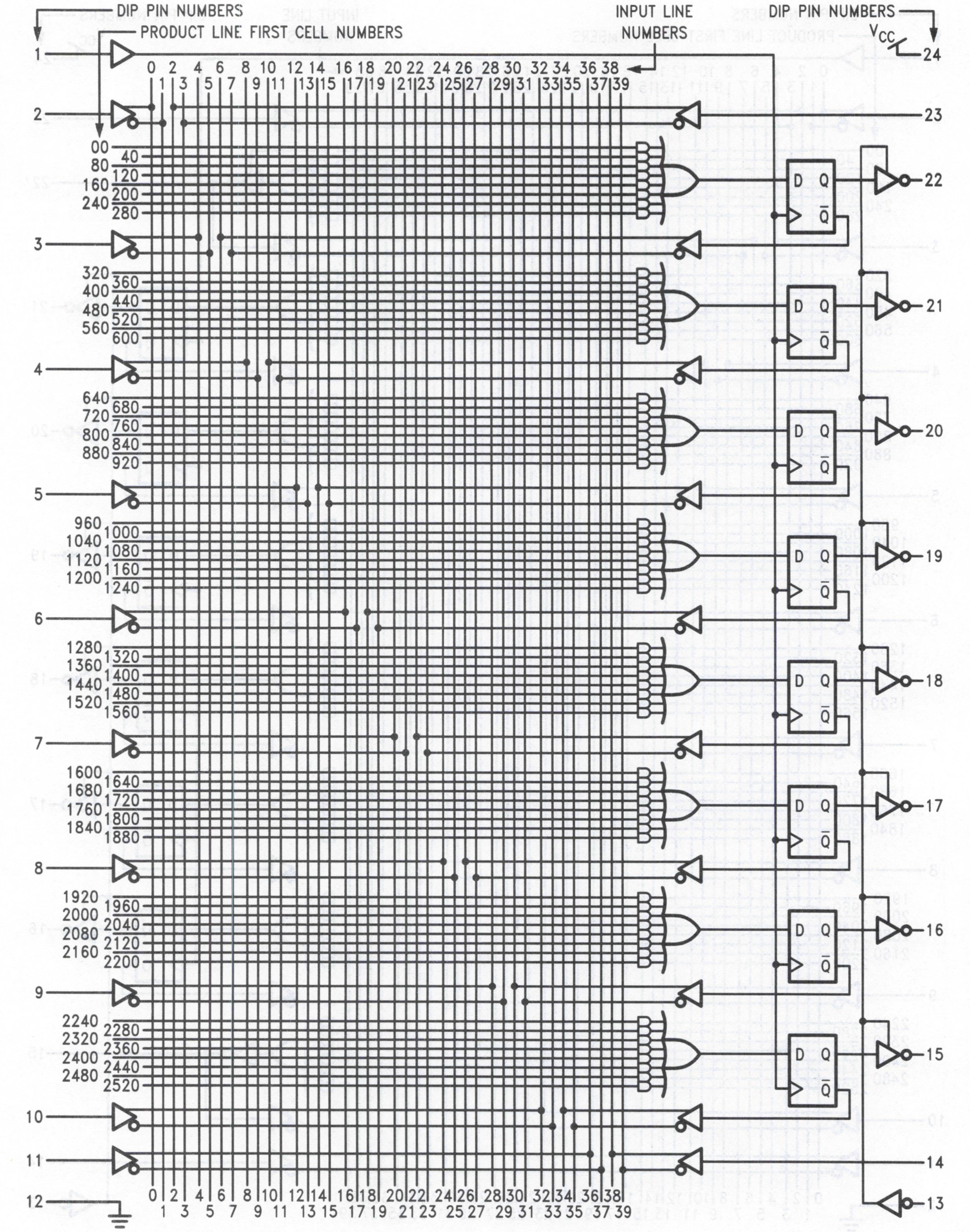
JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

Logic Diagram—PAL20R6


DIP PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
24
1
2
3
4
5
6
7
8
9
10
11
12
23
22
21
20
19
18
17
16
15
14
13
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
00 40 80 120 160 200 240 280
320 360 400 440 480 520 560 600
640 680 720 760 800 840 880 920
960 1000 1040 1080 1120 1160 1200 1240
1280 1320 1360 1400 1440 1480 1520 1560
1600 1640 1680 1720 1760 1800 1840 1880
1920 1960 2000 2040 2080 2120 2160 2200
2240 2280 2320 2360 2400 2440 2480 2520
D
Q
Q̄
TL/L/9394-21

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

Logic Diagram—PAL20R8


DIP PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000
2040
2080
2120
2160
2200
2240
2280
2320
2360
2400
2440
2480
2520
D
Q
Q̄

TL/L/9394-22

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

National Semiconductor

Programmable Array Logic (PAL®) 24-Pin Polarity PAL Famiy—Series B

General Description

The PAL family utilizes National Semiconductor's advanced oxide isolated Schottky TTL process and bipolar PROM fusible-link technology to provide user-programmable logic to replace conventional SSI/MSI gates and flip-flops. Typical chip count reduction gained by using PAL devices is greater than 4:1.

The family lets the systems engineer customize his chip by opening fusible links to configure AND and OR gates to perform his desired logic functions. Complex interconnections that previously required time-consuming layout are thus transferred from PC board to silicon where they can be easily modified during prototype checkout or production.

The PAL transfer function is the familiar sum of products with a single array of fusible links. Unlike the PROM, the PAL is a programmable AND array driving a fixed OR array. (The PROM is a fixed AND array driving a programmable OR array). In addition, the PAL family offers the options of having variable input/output ratios, programmable TRI-STATE® outputs and having registers with feedback.

The programmable output polarity feature allows the user to program individual outputs either active high or active low. This feature eliminates any possible need for inversion of signals outside the device.

Registers consist of D-type flip-flops that are loaded on the low-to-high transition of the clock. The registers power up with a high (V_{OH}) at the output pin, regardless of the polarity fuse.

The entire PAL family is programmed on inexpensive conventional programmers with appropriate personality and socket adapter cards. Once the PAL is programmed and verified, two additional fuses may be blown to defeat verification. This feature gives the user a proprietary circuit which is very difficult to copy.

Features

- 15 ns maximum propagation delay (combinatorial)
- User-programmable replacement for TTL logic
- Large variety of JEDEC-compatible programming equipment and design development software available
- Fully supported by National PLAN™ development software
- Power-up reset for registered outputs
- Register preload facilitates device testing
- Security fuse prevents direct copying of logic patterns
- Skinny DIP packages

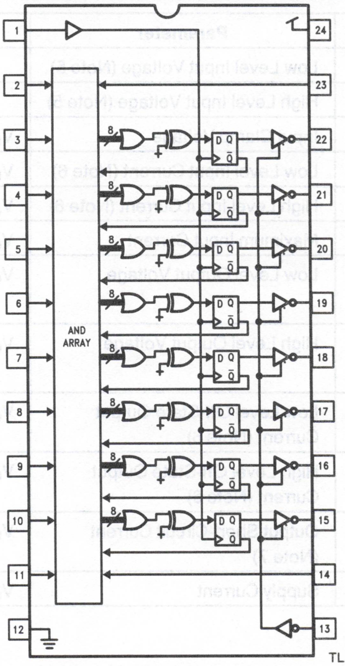
Device Types

Part Number	Dedicated Inputs	Registered Outputs (With Feedback)	Combinatorial	
			I/Os	Outputs
PAL20P8B	14	—	6	2
PAL20RP4B	12	4	4	—
PAL20RP6B	12	6	2	—
PAL20RP8B	12	8	—	—

Speed/Power Versions

Series	Example	Commercial		Military	
		t_{PD}	I_{CC}	t_{PD}	I_{CC}
B	PAL 20P8B	15 ns	210 mA	20 ns	210 mA

Block Diagram—PAL20RP8B



TL/L/9046-35

Series-B (PAL20P8B, PAL20RP4B, PAL20RP6B, PAL20RP8B)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−1.5V to +5.5V
Off-State Output Voltage (Note 2)	−1.5V to +5.5V
Input Current (Note 2)	−30 mA to +5.0 mA
Output Current (I_{OL})	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
ESD Tolerance	2000V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Clock Pulse Width	Low	12	5		10	5		ns
		High	12	5		10	5		ns
t_{SU}	Setup Time from Input or Feedback to Clock		20	10		15	10		ns
t_H	Hold Time of Input after Clock		0	−8		0	−8		ns
f_{CLK}	Clock Frequency (Note 3)	With Feedback		55.6	28.6		55.6	37	MHz
		Without Feedback		100	41.7		100	50	MHz

Electrical Characteristics Over Recommended Operating Conditions (Note 4)

Symbol	Parameter	Test Conditions			Min	Typ	Max	Units
V_{IL}	Low Level Input Voltage (Note 5)						0.8	V
V_{IH}	High Level Input Voltage (Note 5)				2			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I = −18 mA				−0.8	−1.5	V
I_{IL}	Low Level Input Current (Note 6)	V_{CC} = Max, V_I = 0.4V				−0.04	−0.25	mA
I_{IH}	High Level Input Current (Note 6)	V_{CC} = Max, V_I = 2.4V					25	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V					1.0	mA
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 12 mA	MIL		0.3	0.5	V
			I_{OL} = 24 mA	COM				
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −2 mA	MIL	2.4	3.4		V
			I_{OH} = −3.2 mA	COM				
I_{OZL}	Low Level Off-State Output Current (Note 6)	V_{CC} = Max	V_O = 0.4V				−100	µA
I_{OZH}	High Level Off-State Output Current (Note 6)	V_{CC} = Max	V_O = 2.4V				100	µA
I_{OS}	Output Short-Circuit Current (Note 7)	V_{CC} = 5V, V_O = 0V			−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max, Outputs Open				140	210	mA

Series-B (PAL20P8B, PAL20RP4B, PAL20RP6B, PAL20RP8B) (Continued)

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 4: All typical values are for $V_{CC} = 5.0V$ and $T_A = 25°C$.

Note 5: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

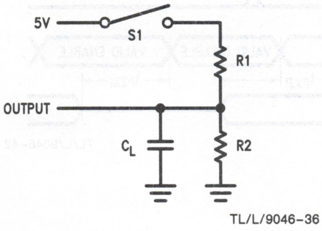
Note 6: Leakage current for bidirectional I/O pins is the worst case between I_{IL} and I_{OZL} or between I_{IH} and I_{OZH}.

Note 7: To avoid invalid readings in other parameter tests it is preferable to conduct the I_{OS} test last. To minimize internal heating, only one output should be shorted at a time with a maximum duration of 1.0 sec. each. Prolonged shorting of a high output may raise the chip temperature above normal and permanent damage may result.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Military			Commerical			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	C_L = 50 pF, S1 Closed		11	20		11	15	ns
t_{CLK}	Clock to Registered Output or Feedback	C_L = 50 pF, S1 Closed		8	15		8	12	ns
t_{PZXG}	$\overline{G}$ Pin to Registered Output Enabled	C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		10	20		10	15	ns
t_{PXZG}	$\overline{G}$ Pin to Registered Output Disabled	C_L = 5 pF, from V_{OH}: S1 Open, from V_{OL}: S1 Closed		11	20		11	15	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		10	20		10	15	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	C_L = 5 pF, from V_{OH}: S1 Open, from V_{OL}: S1 Closed		11	20		11	15	ns
t_{RESET}	Power-Up to Registered Output High			600	1000		600	1000	ns

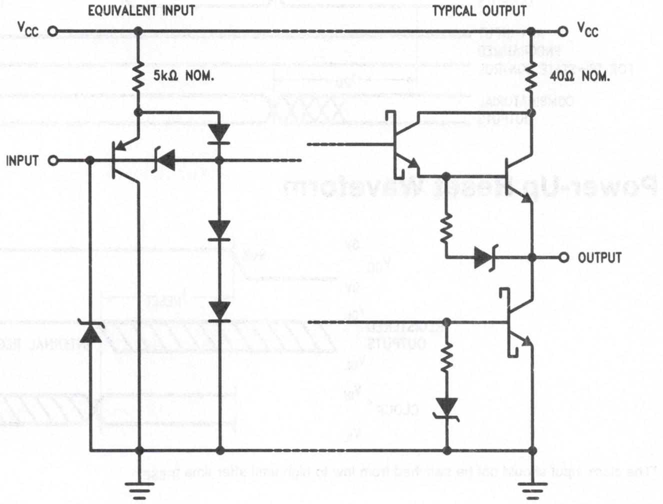
Test Load



TL/L/9046-36

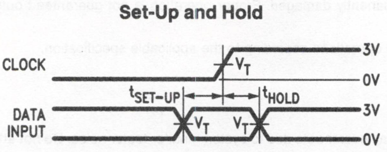
MIL	COM'L
R1 = 390	R1 = 200
R2 = 750	R2 = 390

Schematic of Inputs and Outputs

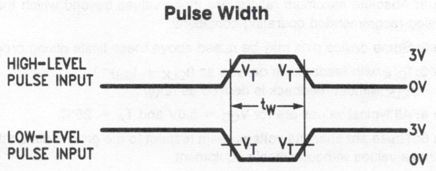


TL/L/9046-37

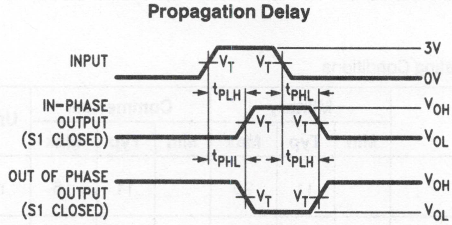
Test Waveforms


Set-Up and Hold
CLOCK
3V
V_T
0V
t_{SET-UP}
t_{HOLD}
DATA INPUT
3V
V_T
V_T
0V

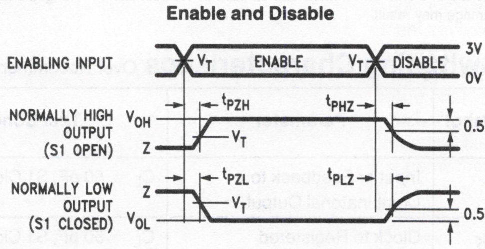
TL/L/9046-38


Pulse Width
HIGH-LEVEL PULSE INPUT
3V
V_T
V_T
0V
t_W
LOW-LEVEL PULSE INPUT
3V
V_T
V_T
0V

TL/L/9046-39


Propagation Delay
INPUT
V_T
V_T
3V
0V
t_{PLH}
t_{PHL}
IN-PHASE OUTPUT (S1 CLOSED)
V_T
V_T
V_{OH}
V_{OL}
t_{PHL}
t_{PLH}
OUT OF PHASE OUTPUT (S1 CLOSED)
V_T
V_T
V_{OH}
V_{OL}

TL/L/9046-40


Enable and Disable
ENABLING INPUT
V_T
ENABLE
V_T
DISABLE
3V
0V
t_{PZH}
t_{PHZ}
NORMALLY HIGH OUTPUT (S1 OPEN)
V_{OH}
Z
V_T
0.5V
t_{PZL}
t_{PLZ}
NORMALLY LOW OUTPUT (S1 CLOSED)
Z
V_{OL}
V_T
0.5V

TL/L/9046-41

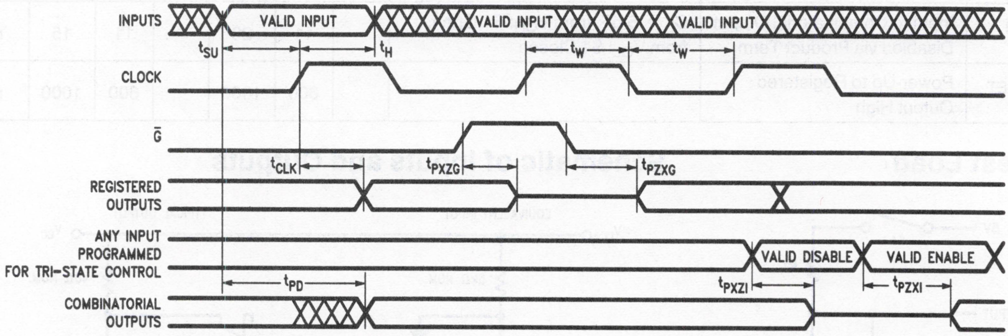
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

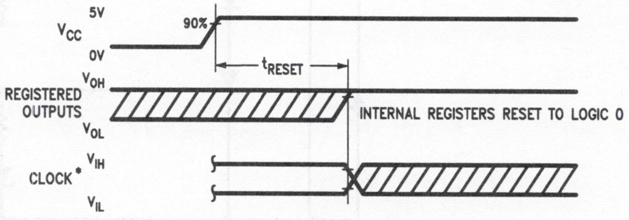
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/9046-42

Power-Up Reset Waveform


5V
V_{CC}
0V
90%
t_{RESET}
V_{OH}
REGISTERED OUTPUTS
V_{OL}
INTERNAL REGISTERS RESET TO LOGIC 0
V_{IH}
CLOCK *
V_{IL}

TL/L/9046-43

*The clock input should not be switched from low to high until after time t_{RESET}.

Functional Description

All of the 24-pin Polarity PAL logic arrays consist of 20 complementary input lines and 64 product-term lines with a programmable fuse link at each intersection (2560 fuses). The product terms are organized into eight groups of eight each. Seven or eight of the product terms in each group connect into an OR-gate to produce the sum-of-products logic function, depending on whether the output is combinatorial or registered.

An unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses) are in the high logic state. Therefore, if both the true and complement of at least one array input is left connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed device). Conversely, if all fuses on a product term were programmed, the product term and the resulting logic function would be held in the high state.

The Polarity PAL family consists of four device types with differing mixtures of combinatorial and registered outputs. The 20P8, 20RP4, 20RP6 and 20RP8 architectures have 0, 4, 6 and 8 registered outputs, respectively, with the balance of the 8 outputs combinatorial. All outputs have TRI-STATE capability.

Each combinatorial output has a seven product-term logic function, with the eighth product term being used for TRI-STATE control. A combinatorial output is enabled while the TRI-STATE product term is satisfied (true). Combinatorial outputs also have feedback paths from the device pins into the logic array (except for two outputs on the 20P8). This allows a pin to perform bidirectional I/O or, if the associated logic function were left unprogrammed, the output driver would remain disabled and the pin could be used as an additional dedicated input.

Registered outputs each have an eight product-term logic function feeding into a D-type flip-flop. All registers are triggered by the high-going edge of the clock input pin. All registered outputs are controlled by a common output enable ($\overline{G}$) pin (enabled while low). The output of each register is also fed back into the logic array via an internal path. This provides for sequential logic circuits (state machines, counters, etc.) which can be sequenced even while the outputs are disabled.

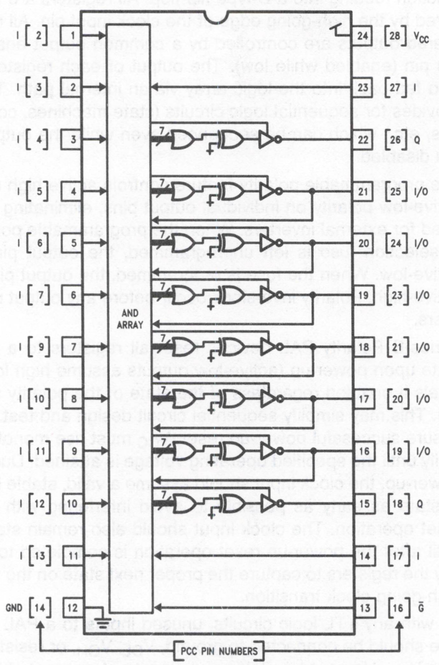
The programmable polarity feature controls active-high and active-low polarity on individual output pins, eliminating the need for external inverters. When the programmable polarity selection fuse is left unprogrammed, the output pin is active-low. When the fuse is programmed, the output pin is active-high. Polarity inversions occur before any output registers.

Series-B Polarity PAL devices reset all registers to a low state upon power-up (active-low outputs assume high logic levels if enabled regardless of the state of the polarity fuses). This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

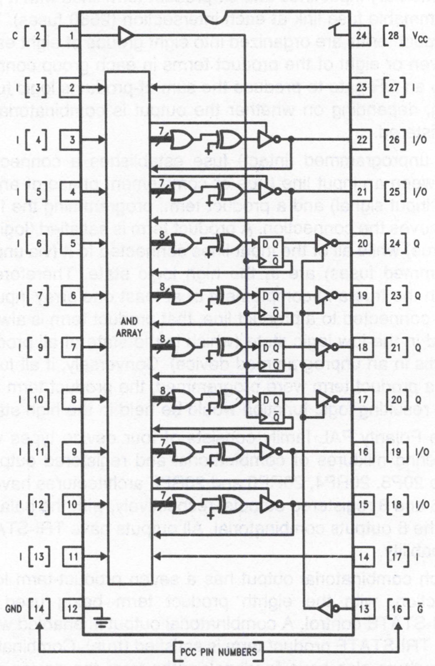
24-Pin Polarity PAL Family Block Diagrams—DIP Connections

PAL20P8


I [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
AND ARRAY
7
24 [28] VCC
23 [27] I
22 [26] Q
21 [25] I/O
20 [24] I/O
19 [23] I/O
18 [21] I/O
17 [20] I/O
16 [19] I/O
15 [18] Q
14 [17] I
13 [16] I
PCC PIN NUMBERS

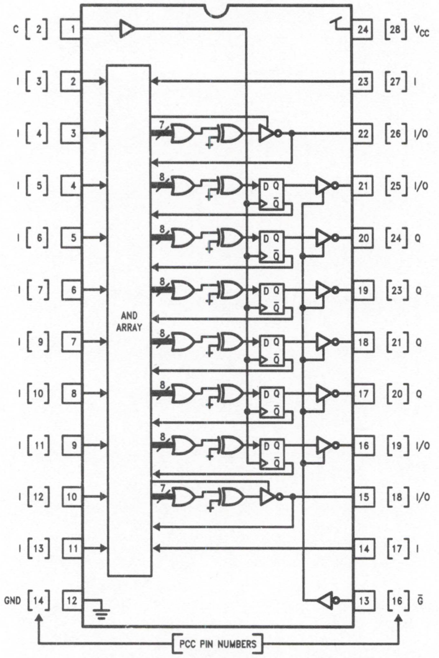
TL/L/9046-44

PAL20RP4


C [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
AND ARRAY
7
8
D Q
Q̅
24 [28] VCC
23 [27] I
22 [26] I/O
21 [25] I/O
20 [24] Q
19 [23] Q
18 [21] Q
17 [20] Q
16 [19] I/O
15 [18] I/O
14 [17]
13 [16] G̅
PCC PIN NUMBERS

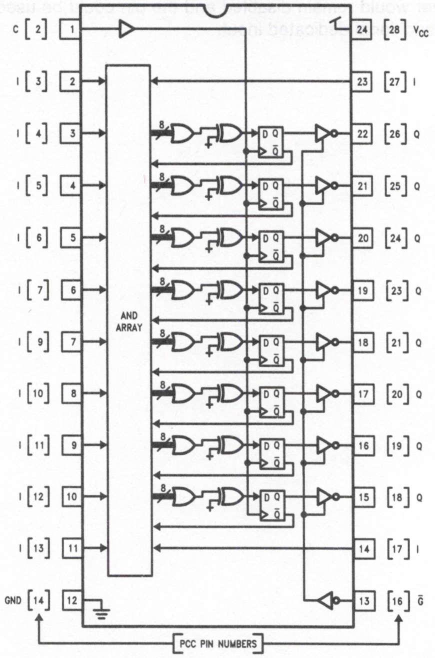
TL/L/9046-45

PAL20RP6


C [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
AND ARRAY
7
8
D Q
Q̅
24 [28] VCC
23 [27] I
22 [26] I/O
21 [25] I/O
20 [24] Q
19 [23] Q
18 [21] Q
17 [20] Q
16 [19] I/O
15 [18] I/O
14 [17] I
13 [16] G̅
PCC PIN NUMBERS

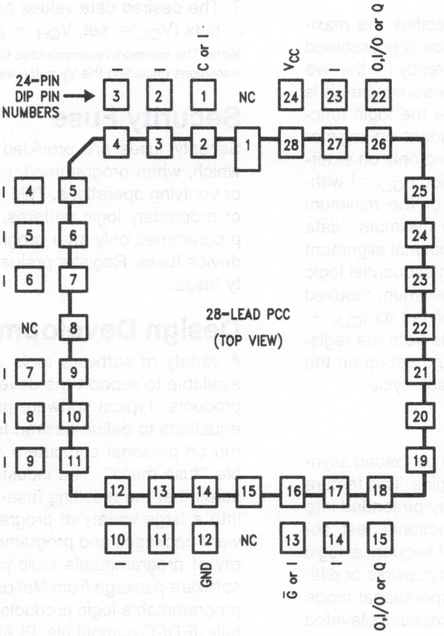
TL/L/9046-46

PAL20RP8


C [2] 1
I [3] 2
I [4] 3
I [5] 4
I [6] 5
I [7] 6
I [9] 7
I [10] 8
I [11] 9
I [12] 10
I [13] 11
GND [14] 12
AND ARRAY
8
D Q
Q̅
24 [28] VCC
23 [27] I
22 [26] Q
21 [25] Q
20 [24] Q
19 [23] Q
18 [21] Q
17 [20] Q
16 [19] Q
15 [18] Q
14 [17] I
13 [16] G̅
PCC PIN NUMBERS

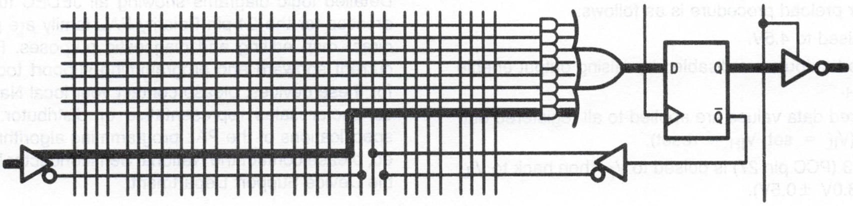
TL/L/9046-47

28-Lead PCC Connection Conversion Diagram


24-PIN
DIP PIN
NUMBERS
C or I
V_{CC}
O,I/O or Q
3
2
1
NC
24
23
22
4
3
2
1
28
27
26
I
4
5
25
21
I/O or Q
I
5
6
24
20
I/O or Q
I
6
7
23
19
I/O or Q
NC
8
28-LEAD PCC
(TOP VIEW)
22
NC
I
7
9
21
18
I/O or Q
I
8
10
20
17
I/O or Q
I
9
11
19
16
I/O or Q
12
13
14
15
16
17
18
10
11
12
NC
13
14
15
GND
G or I
O,I/O or Q

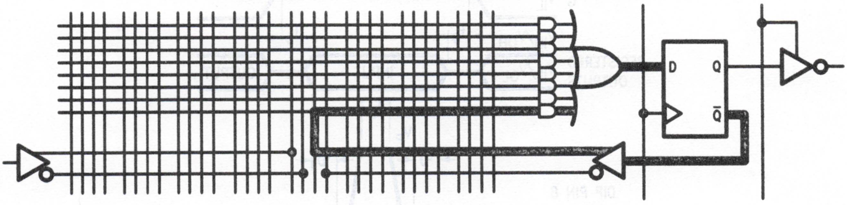
TL/L/9046-48

Typical Registered Logic Function Without Feedback



TL/L/9046-49

Typical Registered Logic Function With Feedback



TL/L/9046-50

Functional Description (Continued)

CLOCK FREQUENCY SPECIFICATION

The clock frequency (f_{CLK}) parameter specifies the maximum speed at which a registered PAL device is guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, where the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e., based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_W high + t_W low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period (f_{CLK}^{-1} with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set-up on the inputs to the registers before the end of each cycle.

Output Register Preload

The preload function allows the registers to be loaded asynchronously from data placed on the output pins. This feature simplifies device testing since any state may be loaded into the registers at any time during the functional test sequence. This allows complete verification of sequential logic circuits, including states that are normally impossible or difficult to reach. Register preload is not an operational mode and is not intended for board level testing because elevated voltage levels are required. The programming system normally provides the preload capability as part of its functional test facility.

The register preload procedure is as follows:

1. V_{CC} is raised to 4.5V.
2. Registered outputs are disabled by raising output enable ($\overline{G}$) to V_{IH}.
3. The desired data values are applied to all registered output pins (V_{IL} = set, V_{IH} = reset).
4. DIP pin 23 (PCC pin 27) is pulsed to V_P, then back to V_{IL}. (V_P = 18.0V ±0.5V).
5. Data inputs are removed from registered output pins.
6. $\overline{G}$ is lowered to V_{IL} to enable registered outputs.
7. The desired data values are verified at all registered outputs (V_{OL} = set, V_{OH} = reset).

Note: The minimum recommended time delay (t_D) between successive input transitions (including the V_P pulse width on DIP pin 8) is 100 ns.

Security Fuse

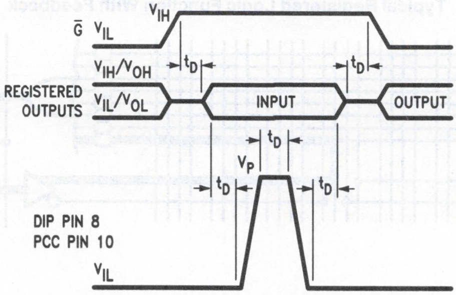
Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses. Register preload is not affected by the security fuses.

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a large variety of programming equipment. Many software packages and programming units support a large variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

Detailed logic diagrams showing all JEDEC fuse-map addresses for the 24-pin Polarity PAL family are provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

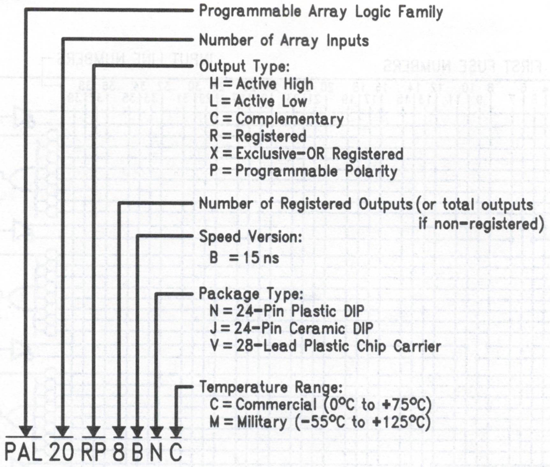
Register Preload Waveform



TL/L/9046-51

Note: V_P = 18.0V ±0.5V, t_D min. = 100 ns

Ordering Information



Programmable Array Logic Family

Number of Array Inputs

Output Type:
- H = Active High
- L = Active Low
- C = Complementary
- R = Registered
- X = Exclusive-OR Registered
- P = Programmable Polarity

Number of Registered Outputs (or total outputs if non-registered)

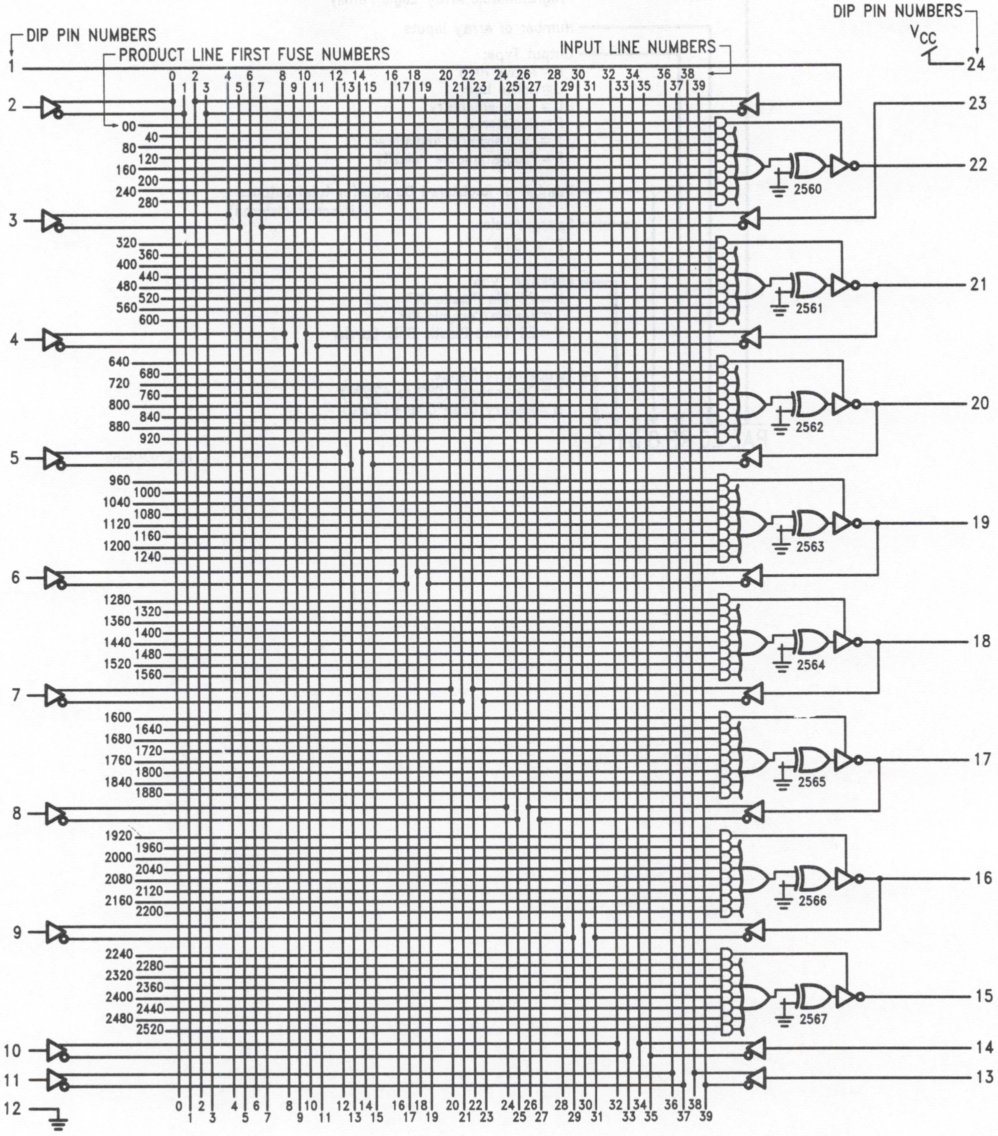
Speed Version:
- B = 15 ns

Package Type:
- N = 24-Pin Plastic DIP
- J = 24-Pin Ceramic DIP
- V = 28-Lead Plastic Chip Carrier

Temperature Range:
- C = Commercial (0°C to +75°C)
- M = Military (−55°C to +125°C)

TL/L/9046-52

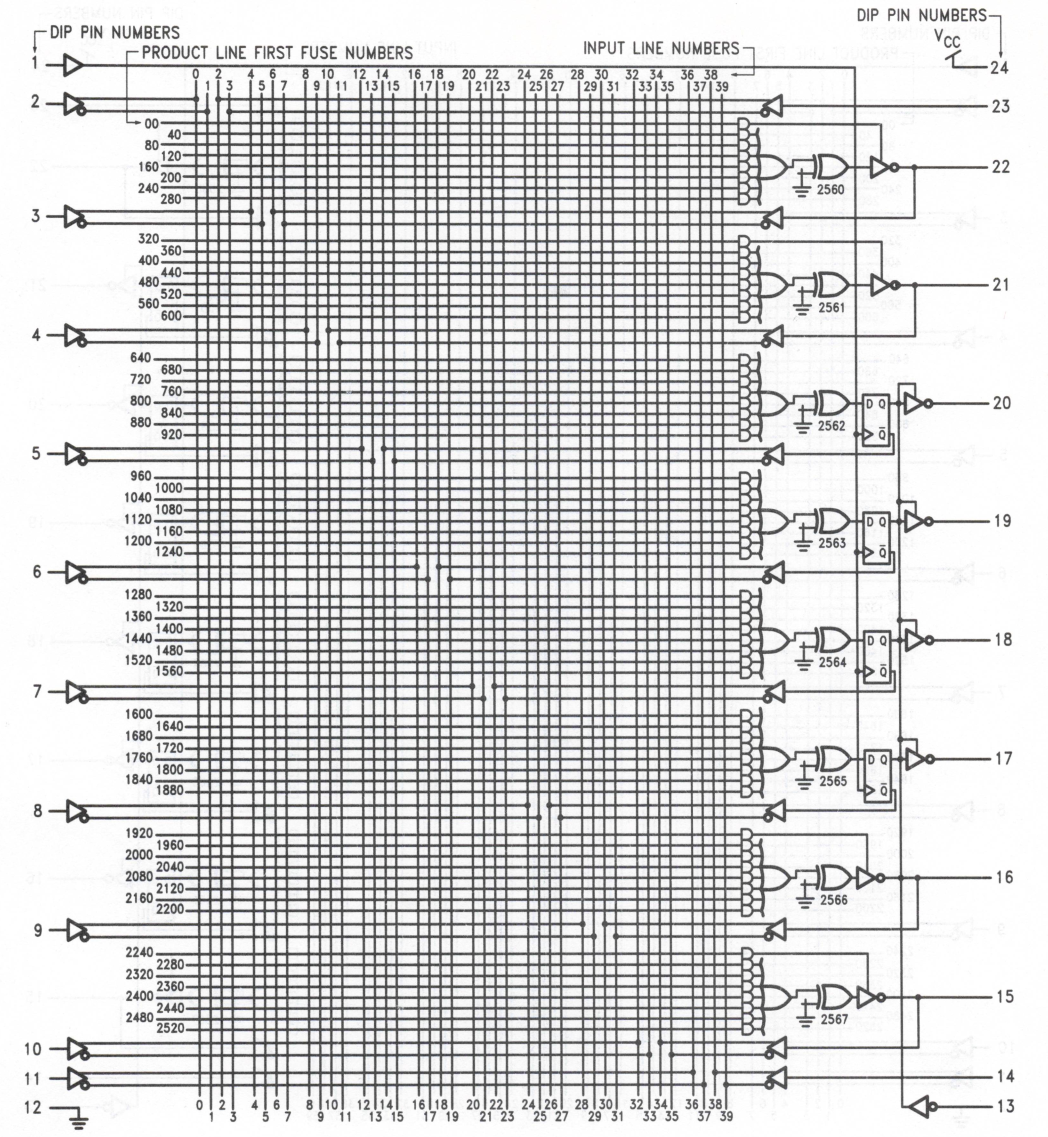
Logic Diagram—PAL20P8B


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000
2040
2080
2120
2160
2200
2240
2280
2320
2360
2400
2440
2480
2520
2560
2561
2562
2563
2564
2565
2566
2567

TL/L/9046-7

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

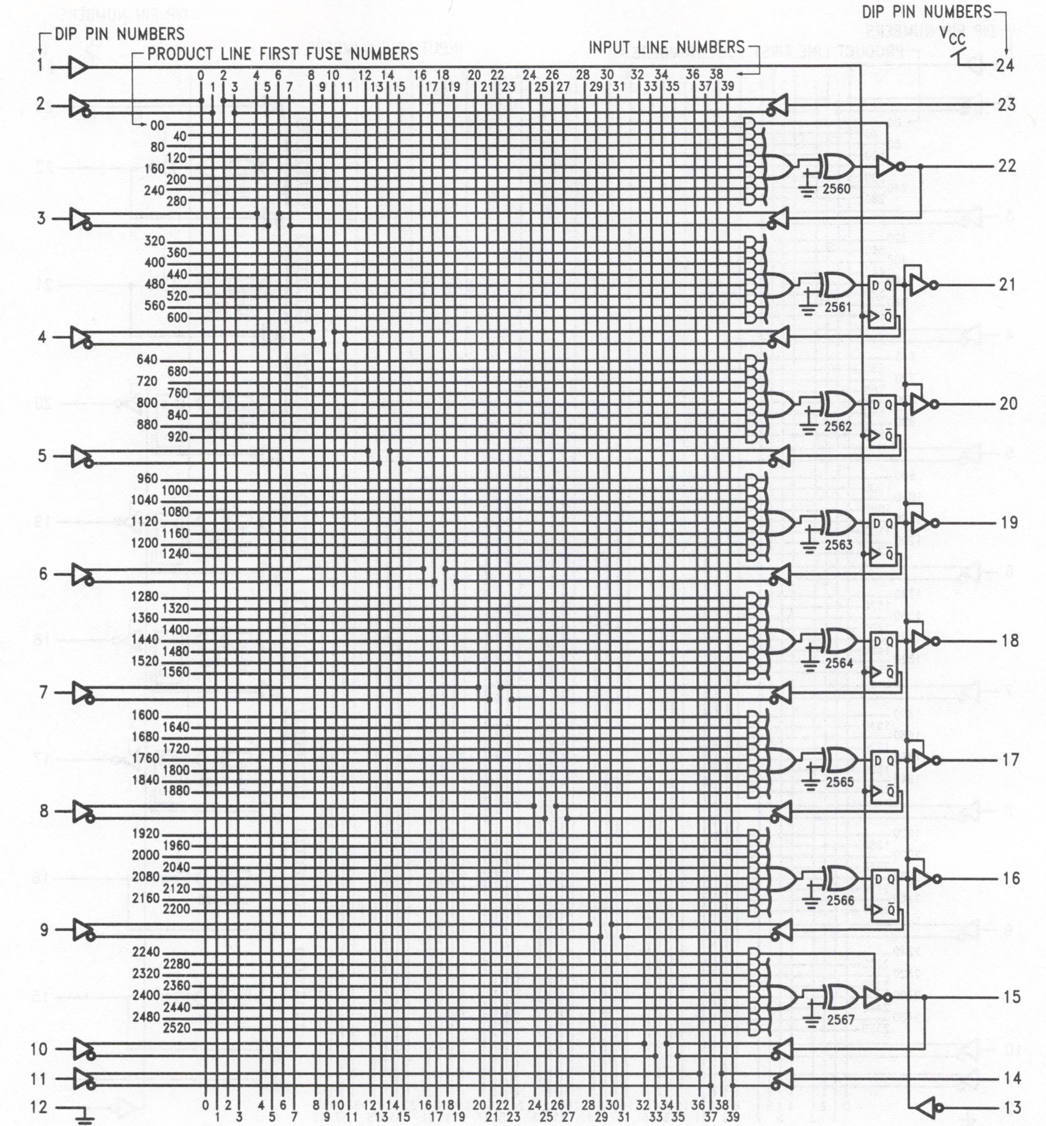
Logic Diagram—PAL20RP4B


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000
2040
2080
2120
2160
2200
2240
2280
2320
2360
2400
2440
2480
2520
2560
2561
2562
2563
2564
2565
2566
2567
D Q
Q̄

TL/L/9046-8

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

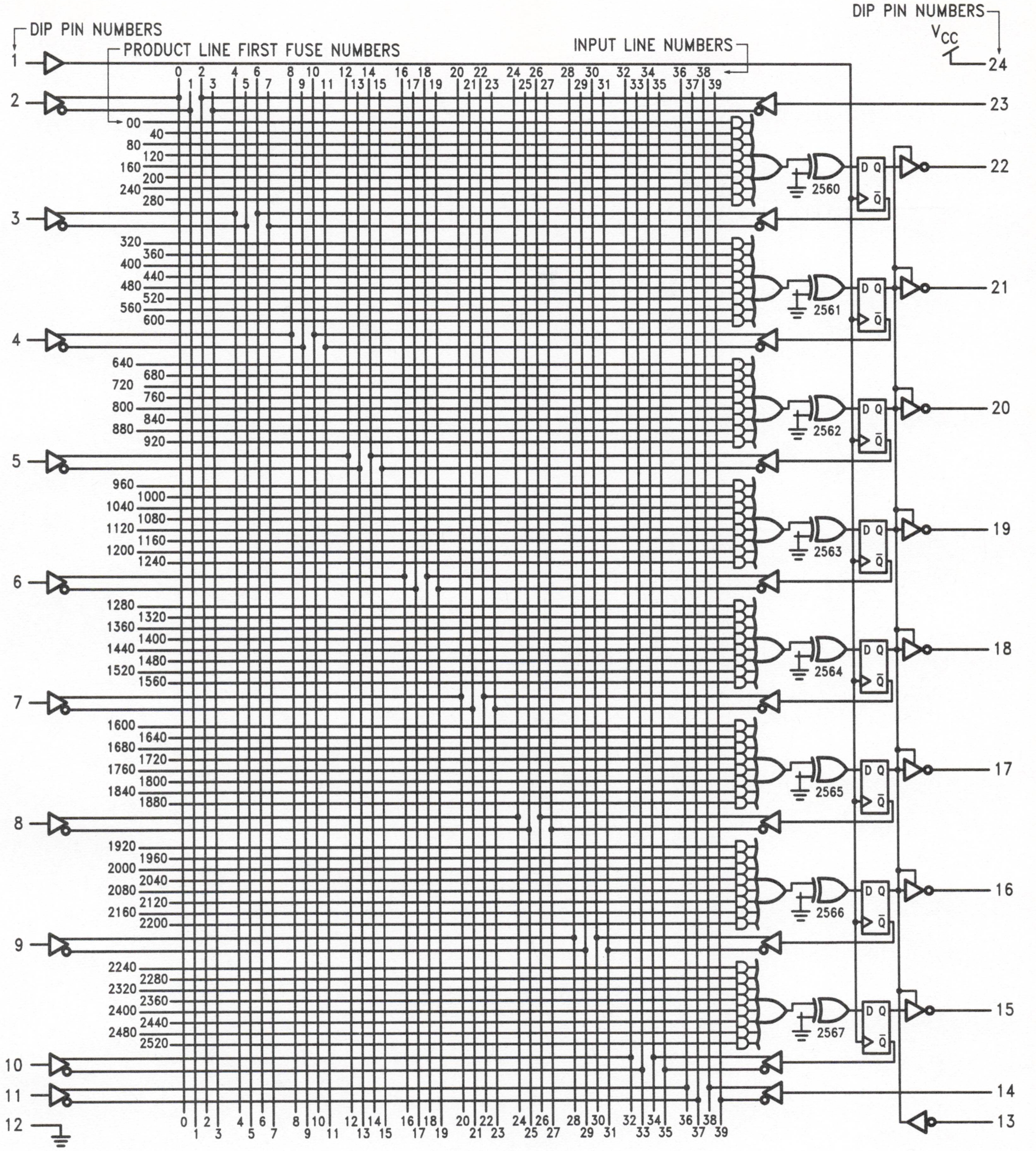
Logic Diagram—PAL20RP6B


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000
2040
2080
2120
2160
2200
2240
2280
2320
2360
2400
2440
2480
2520
2560
2561
2562
2563
2564
2565
2566
2567
D Q
Q̄

TL/L/9046-9

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

Logic Diagram—PAL20RP8B


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
VCC
1
2
3
4
5
6
7
8
9
10
11
12
24
23
22
21
20
19
18
17
16
15
14
13
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
00
40
80
120
160
200
240
280
320
360
400
440
480
520
560
600
640
680
720
760
800
840
880
920
960
1000
1040
1080
1120
1160
1200
1240
1280
1320
1360
1400
1440
1480
1520
1560
1600
1640
1680
1720
1760
1800
1840
1880
1920
1960
2000
2040
2080
2120
2160
2200
2240
2280
2320
2360
2400
2440
2480
2520
2560
2561
2562
2563
2564
2565
2566
2567
D Q
Q̄

TL/L/9046-10

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

National Semiconductor

Programmable Array Logic (PAL®) PAL16RA8

General Description

The PAL16RA8 is a new member of National's broad PAL® family. It provides several new features which will dramatically benefit PAL users. National Semiconductor's advanced Schottky TTL process with titanium tungsten fusible links is used in manufacturing the RA (Registered Asynchronous) devices.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming the programmable cells to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

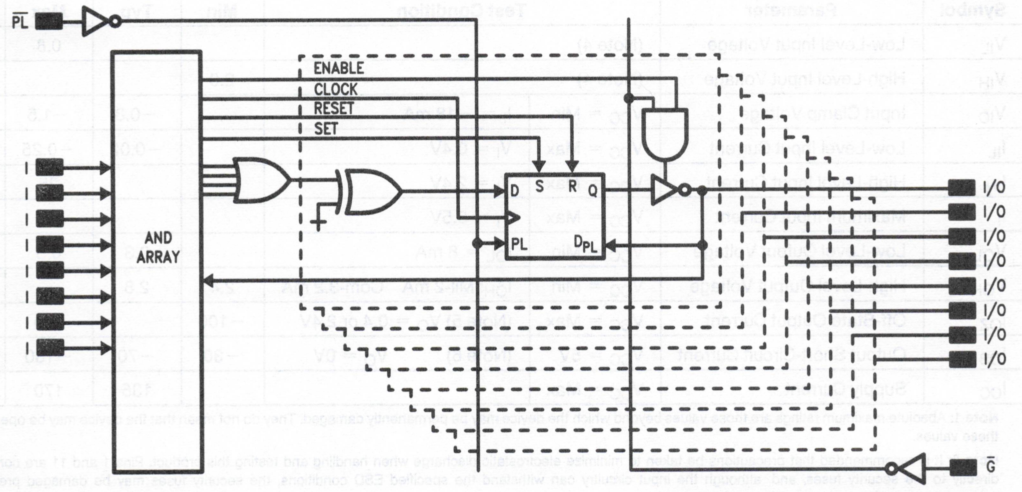
The PAL16RA8 is made up of eight Output Logic Macro Cells (OLMC). Four AND array outputs feed into the fixed OR-gate for each OLMC to generate the device's output functions. Four other AND array outputs are used for control functions in the OLMC. With a robust mixture of logic derived controlled functions and selectable output data paths, the PAL16RA8 provides an ideal solution for registered random logic applications.

This device is housed in a 20-pin 300 mil DIP. A 20-pin PCC package is also available. It can be programmed by most PAL programmers.

Features

- Programmable asynchronous set and reset
- Individually programmable clocks
- Programmable and hard-wired TRI-STATE® outputs
- Programmable output polarity
- Registers can be bypassed individually
- Register preload guarantees testability
- Outputs can be reconfigured as inputs
- Power-up reset for registered outputs
- Fully supported by National PLAN™ development software
- A variety of JEDEC-compatible programming equipment and design development software available
- Security fuse prevents direct copying of logic patterns

Block Diagram—PAL16RA8



TL/L/9253-18

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Operating	Programming
Supply Voltage V_{CC}	7.0V	12.0V
Input Voltage	5.5V	22.0V

	Operating	Programming
Off-State Output Voltage	5.5V	12.0V
Storage Temperature		−65°C to +150°C
ESD Tolerance (Note 2)		1000V

$C_{ZAP} = 100\ \text{pF}$
$R_{ZAP} = 1500\Omega$
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military			Commercial			Units
			Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Pulse Width of Clocking Input (High/Low)		25	13		20	13		ns
t_{WP}	Pulse Width of Preload ($\overline{PL}$) Input (Low)		45	15		35	15		ns
t_{SU}	Setup Time from Input or Feedback to Clocking Input		25	10		20	10		ns
t_{SUP}	Setup Time from Input to $\overline{PL}$ High		30	5		25	5		ns
t_H	Hold Time of Input after Clocking Input	Polarity Fuse Intact	10	−2		10	−2		ns
		Polarity Fuse Blown	0	−6		0	−6		
t_{HP}	Hold Time of Input after $\overline{PL}$ High		30	5		25	5		ns
f_{CLK}	Clock Frequency (Note 3)	With Feedback		37	16.6		37	20	MHz
		Without Feedback		38.5	20		38.5	25	

Electrical Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Condition		Min	Typ	Max	Units
V_{IL}	Low-Level Input Voltage	(Note 4)				0.8	V
V_{IH}	High-Level Input Voltage	(Note 4)		2.0			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min	$I_I = -18$ mA		−0.8	−1.5	V
I_{IL}	Low-Level Input Current	V_{CC} = Max	$V_I = 0.4V$		−0.02	−0.25	mA
I_{IH}	High-Level Input Current	V_{CC} = Max	$V_I = 2.4V$			25	µA
I_I	Maximum Input Current	V_{CC} = Max	$V_I = 5.5V$			1	mA
V_{OL}	Low-Level Output Voltage	V_{CC} = Min	$I_{OL} = 8$ mA		0.3	0.5	V
V_{OH}	High-Level Output Voltage	V_{CC} = Min	I_{OH}: Mil-2 mA Com-3.2 mA	2.4	2.8		V
I_{OZ}	Off-State Output Current	V_{CC} = Max	(Note 5) V_O = 0.4 or 2.4V	−100		100	µA
I_{OS}	Output Short-Circuit Current	V_{CC} = 5V	(Note 6) V_O = 0V	−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max			135	170	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. They do not mean that the device may be operated at these values.

Note 2: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 11 are connected directly to the security fuses, and, although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged preventing subsequent programming and verification operations.

Note 3: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2\ t_W)^{-1}$.

Note 4: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

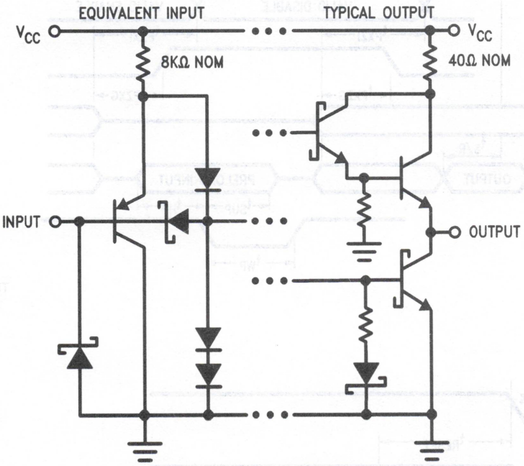
Note 5: I/O leakage as the worst case of I_{OZX} or I_{IX}, e.g. I_{IL} and I_{OZL}.

Note 6: During I_{OS} measurement, only one output at a time should be grounded. Permanant damage otherwise may result.

Switching Characteristics Over Recommended Operating Conditions

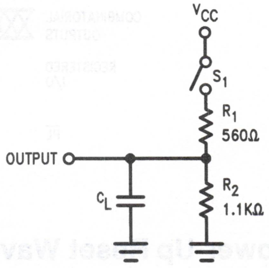
Symbol	Parameter		Test Conditions	Military			Commercial			Units
				Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	Polarity Fuse Intact	$C_L = 50$ pF, S1 Closed		20	35		20	30	ns
		Polarity Fuse Blown			25	40		25	35	
t_{CLK}	Clocking Input to Registered Output or Feedback		$C_L = 50$ pF, S1 Closed	10	17	35	10	17	30	ns
t_{PZXG}	$\overline{G}$ Pin to Output Enabled		$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		10	25		10	20	ns
t_{PZXG}	$\overline{G}$ Pin to Output Disabled		$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		10	25		10	20	ns
t_{PZXI}	Input to Output Enabled via Product Term		$C_L = 50$ pF, Active High: S1 Open, Active Low: S1 Closed		18	35		18	30	ns
t_{PXZI}	Input to Output Disabled via Product Term		$C_L = 5$ pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		15	35		15	30	ns
t_{RESET}	Power-Up to Registered Output-High				600	1000		600	1000	ns
t_S	Asynchronous Set Input to Registered Output Low				22	40		22	35	ns
t_R	Asynchronous Reset Input to Registered Output High				27	45		27	40	ns

Schematic of Inputs and Outputs



TL/L/9253-3

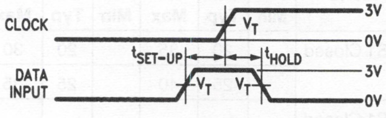
Test Load



TL/L/9253-4

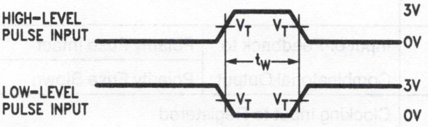
Test Waveforms

Set-Up and Hold


CLOCK
DATA INPUT
V_T
t_{SET-UP}
t_{HOLD}
3V
0V
3V
0V

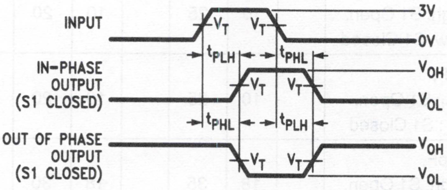
TL/L/9253-19

Pulse Width


HIGH-LEVEL PULSE INPUT
LOW-LEVEL PULSE INPUT
V_T
t_W
3V
0V
3V
0V

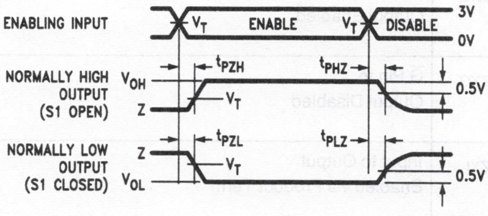
TL/L/9253-20

Propagation Delay


INPUT
IN-PHASE OUTPUT (S1 CLOSED)
OUT OF PHASE OUTPUT (S1 CLOSED)
V_T
t_{PLH}
t_{PHL}
t_{PHL}
t_{PLH}
3V
0V
V_{OH}
V_{OL}
V_{OH}
V_{OL}

TL/L/9253-21

Enable and Disable


ENABLING INPUT
ENABLE
DISABLE
V_T
NORMALLY HIGH OUTPUT (S1 OPEN)
NORMALLY LOW OUTPUT (S1 CLOSED)
V_{OH}
Z
Z
V_{OL}
t_{PZH}
t_{PHZ}
t_{PZL}
t_{PLZ}
3V
0V
0.5V
0.5V

TL/L/9253-22

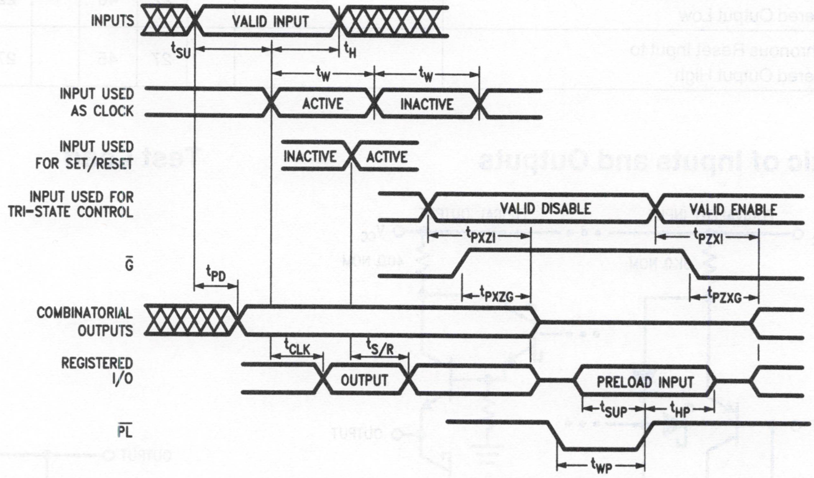
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

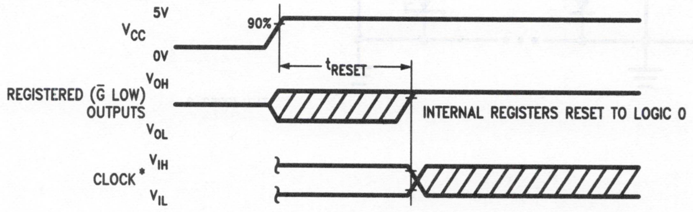
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS
VALID INPUT
t_{SU}
t_H
t_W
t_W
INPUT USED AS CLOCK
ACTIVE
INACTIVE
INPUT USED FOR SET/RESET
INACTIVE
ACTIVE
INPUT USED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PXZI}
t_{PZXI}
$\overline{G}$
t_{PD}
t_{PXZG}
t_{PZXG}
COMBINATORIAL OUTPUTS
t_{CLK}
$t_{S/R}$
REGISTERED I/O
OUTPUT
PRELOAD INPUT
t_{SUP}
t_{HP}
$\overline{PL}$
t_{WP}

TL/L/9253-23

Power-Up Reset Waveform


V_{CC}
5V
0V
90%
t_{RESET}
REGISTERED ($\overline{G}$ LOW) OUTPUTS
V_{OH}
V_{OL}
INTERNAL REGISTERS RESET TO LOGIC 0
CLOCK*
V_{IH}
V_{IL}

TL/L/9253-24

*The clock input should not be switched from low to high until after time t_{RESET}.

Functional Description

The PAL16RA8 logic array consists of 16 complementary input lines and 64 product-term lines with a programmable fuse link at each intersection (2048 fuses). The product terms are organized into eight groups of eight each. Four of the eight product terms in each group connect into an OR-gate to produce the sum-of-products logic function. The remaining four product terms in each group are used for control functions in the Output Logic Macro Cell (OLMC) as shown in *Figure 1*.

An unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses) are in the high logic state. Therefore, if both the true and complement of at least one array input is left connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed device). Conversely, if all fuses on a product term were programmed, the product term and the resulting logic function would be held in the high state.

PROGRAMMABLE SET AND RESET

In each cell, two product lines are dedicated to asynchronous set and reset. If the set product line is high, the register output becomes a logic 1, the output pin becomes a 0. If the reset product line is high, the register output becomes a logic 0, the output pin becomes a 1. The operation of the programmable set and reset overrides the clock.

INDIVIDUALLY PROGRAMMABLE REGISTER BYPASS

If both the set and reset product lines are high, the sum-of-products bypasses the register and appears immediately at the output, thus making the output combinatorial. This allows each output to be configured in the registered or combinatorial mode.

PROGRAMMABLE CLOCK

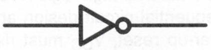
One of the product lines in each group is connected to the clock. This provides the user with the additional flexibility of a programmable clock, so each output can be clocked independently of all the others.

PROGRAMMABLE AND HARD-WIRED TRI-STATE OUTPUTS

The PAL16RA8 provides a product term dedicated to output control. There is also an output control pin (Pin 11). The output is enabled if both the output control pin is low and the output control product term is HIGH. If the output control pin is high all outputs will be disabled or if an output control product term is low, then that output will be disabled.

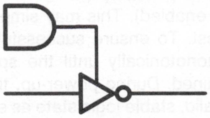
OUTPUT CONTROL ALTERNATIVES

Output Always Enabled



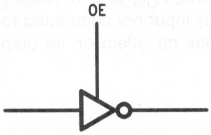
TL/L/9253-6

Programmable



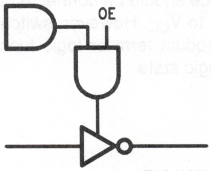
TL/L/9253-7

Hard-Wired



TL/L/9253-8

Combination of Programmable and Hard-Wired

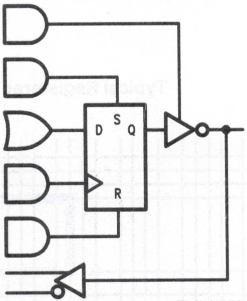


TL/L/9253-9

PROGRAMMABLE OUTPUT POLARITY

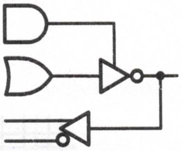
The outputs can be programmed either active-low or active-high. This is represented by the exclusive-or gates shown in the PAL16RA8 Logic Diagram. When the output polarity fuse is blown, the lower input to the exclusive-or gate is high, so the output is active-high. Similarly, when the output polarity fuse is intact, the output is active-low. The programmable output polarity features allows the user a higher degree of flexibility when writing equations.

Registered/Active Low



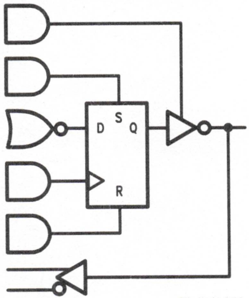
TL/L/9253-10

Combinatorial/Active Low



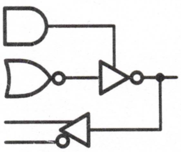
TL/L/9253-11

Registered/Active High



TL/L/9253-12

Combinatorial/Active High



TL/L/9253-13

Functional Description (Continued)

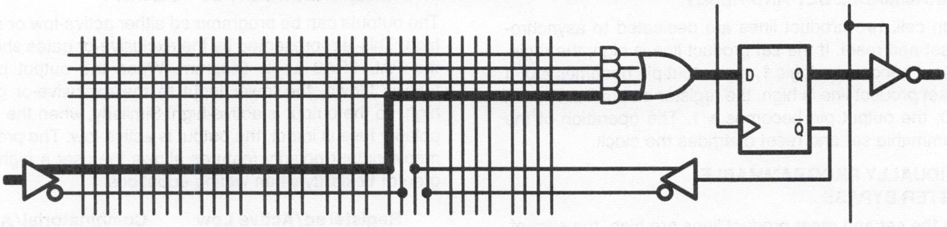
The PAL16RA8 devices reset all registers to a low state upon power-up (active-low outputs assume high logic levels if enabled). This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

CLOCK FREQUENCY SPECIFICATION

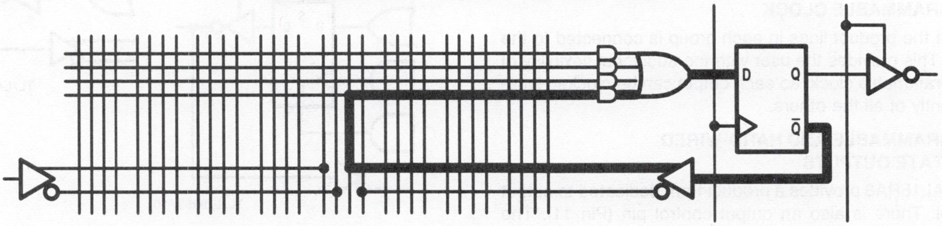
The clock frequency (f_{CLK}) parameter specifies the maximum speed at which a registered PAL device is guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, where the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e.—based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_W high + t_W low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period (f_{CLK}^{-1} with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set-up on the inputs to the registers before the end of each cycle.

Typical Registered Logic Function Without Feedback



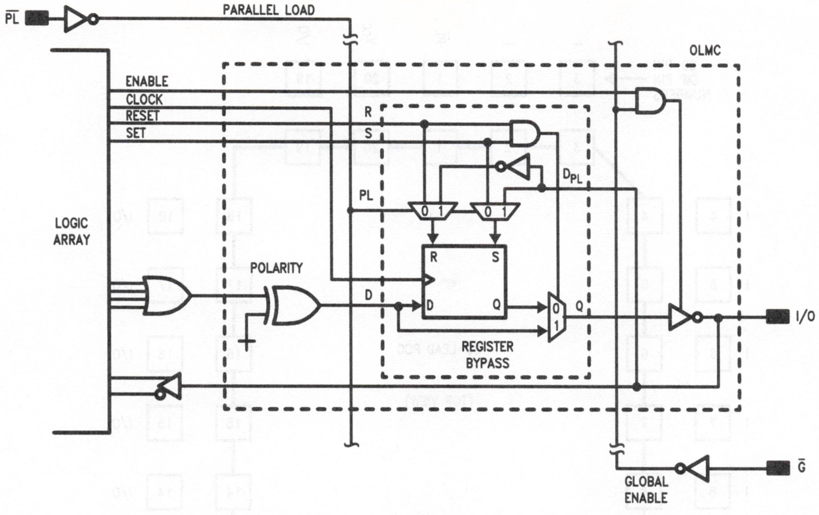
TL/L/9253-27

Typical Registered Logic Function With Feedback



TL/L/9253-28

Functional Description (Continued)

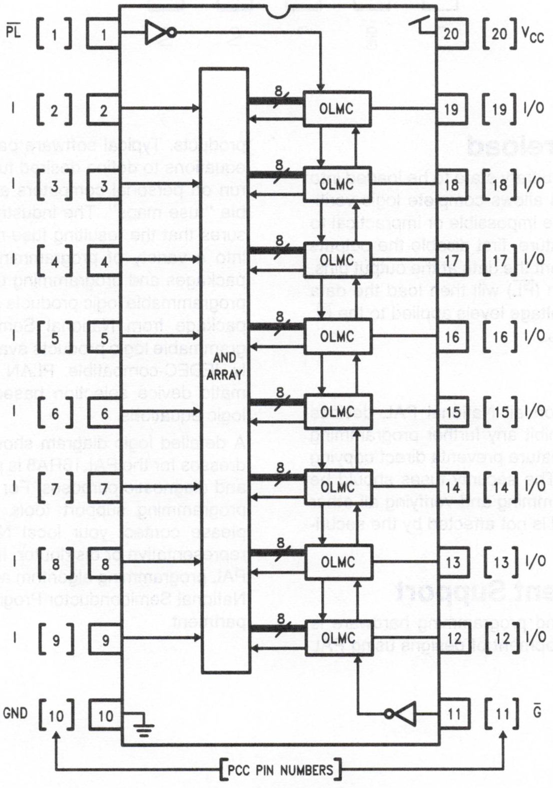


$\overline{PL}$
PARALLEL LOAD
OLMC
ENABLE
CLOCK
RESET
SET
R
S
LOGIC ARRAY
PL
D_{PL}
0 1
0 1
R
S
POLARITY
D
D
Q
Q
0
1
REGISTER BYPASS
I/O
$\overline{G}$
GLOBAL ENABLE

TL/L/9253-25

FIGURE 1. "RA" Output Logic Macrocell Logic Diagram

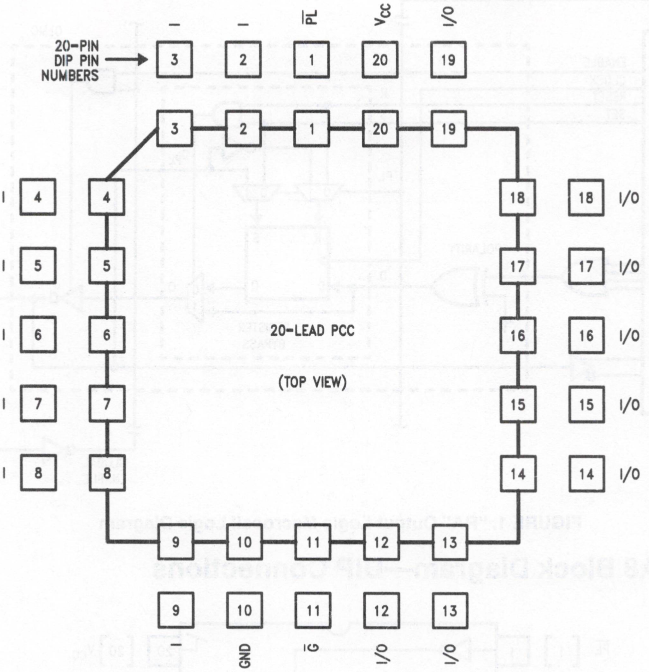
20-Pin PAL16RA8 Block Diagram—DIP Connections



$\overline{PL}$ [1] 1
I [2] 2
I [3] 3
I [4] 4
I [5] 5
I [6] 6
I [7] 7
I [8] 8
I [9] 9
GND [10] 10
20 [20] V_{CC}
19 [19] I/O
18 [18] I/O
17 [17] I/O
16 [16] I/O
15 [15] I/O
14 [14] I/O
13 [13] I/O
12 [12] I/O
11 [11] $\overline{G}$
AND ARRAY
OLMC
8
PCC PIN NUMBERS

TL/L/9253-30

20-Lead PCC Connection Diagram



TL/L/9253-26

Output Register Preload

Register preload allows any arbitrary state to be loaded into the PAL output registers. This allows complete logic verification, including states that are impossible or impractical to reach. To use the preload feature, first disable the outputs by bringing $\overline{OE}$ high, and present the data at the output pins. A low-level on the preload pin ($\overline{PL}$) will then load the data into the registers. Note that voltage levels applied to the $\overline{PL}$ inputs are standard TTL levels.

Security Fuse

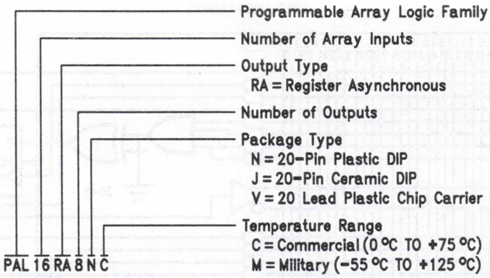
Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses. Register preload is not affected by the security fuses.

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a variety of programming equipment. Many software packages and programming units support a wide variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

A detailed logic diagram showing all JEDEC fuse-map addresses for the PAL16RA8 is provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

Ordering Information



Programmable Array Logic Family

Number of Array Inputs

Output Type
RA = Register Asynchronous

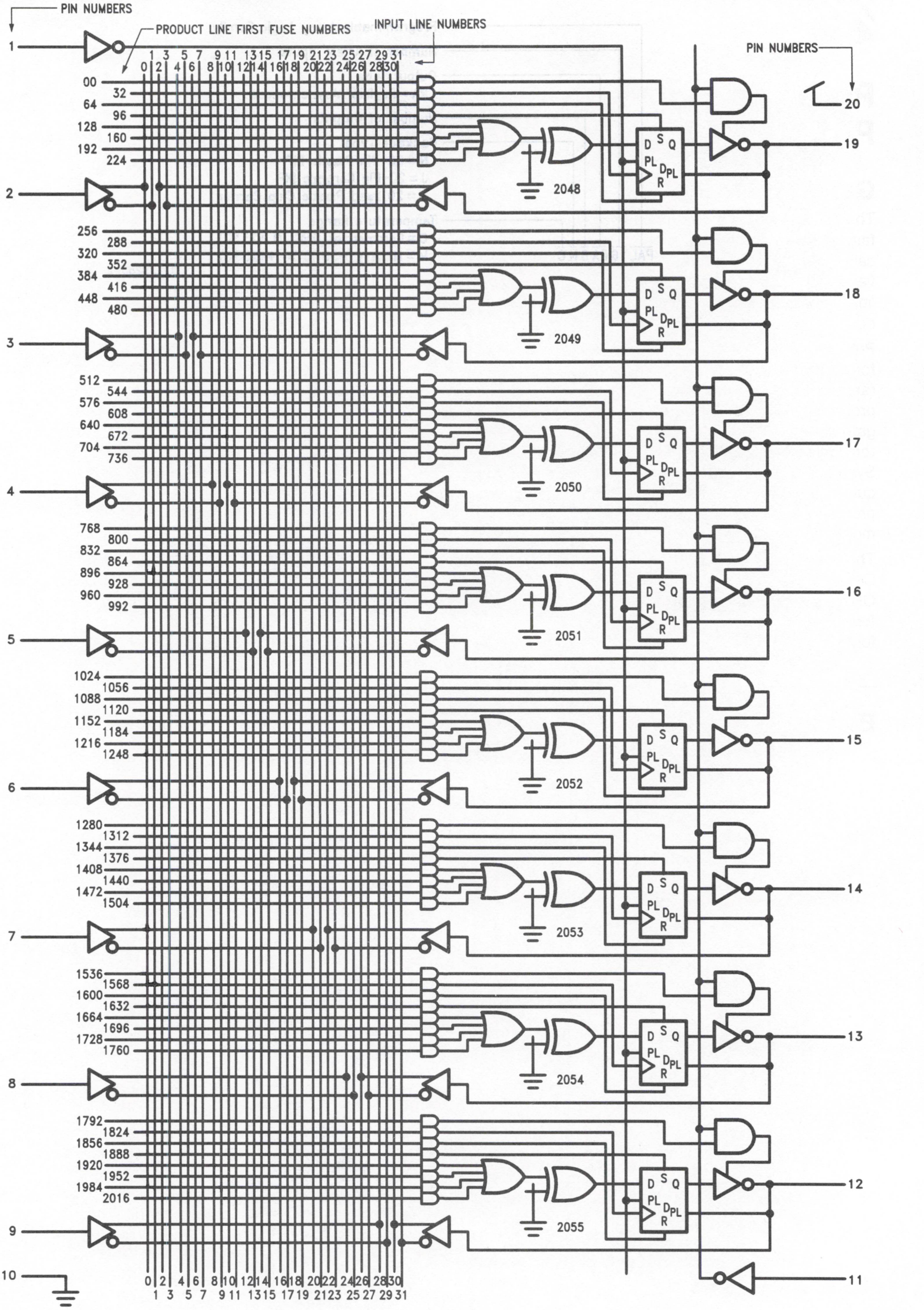
Number of Outputs

Package Type
N = 20-Pin Plastic DIP
J = 20-Pin Ceramic DIP
V = 20 Lead Plastic Chip Carrier

Temperature Range
C = Commercial (0 °C TO +75 °C)
M = Military (−55 °C TO +125 °C)

TL/L/9253-2

Logic Diagram—PAL16RA8


PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
PIN NUMBERS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
00 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
2048
2049
2050
2051
2052
2053
2054
2055
D S Q
PL D_PL
R

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number

TL/L/9253-17

National Semiconductor

Programmable Array Logic (PAL®) PAL20RA10

General Description

The PAL20RA10 is a new member of National's broad PAL family. It provides several new features which will dramatically benefit PAL users. National Semiconductor's advanced Schottky TTL process with titanium tungsten fusible links is used in manufacturing the RA (Registered Asynchronous) devices.

Programmable logic devices provide convenient solutions for a wide variety fo application-specific functions, including random logic, custom decoders, state machines, etc. By programming the programmable cells to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products. A large variety of programming units and software makes design development and functional testing of PAL devices quick and easy.

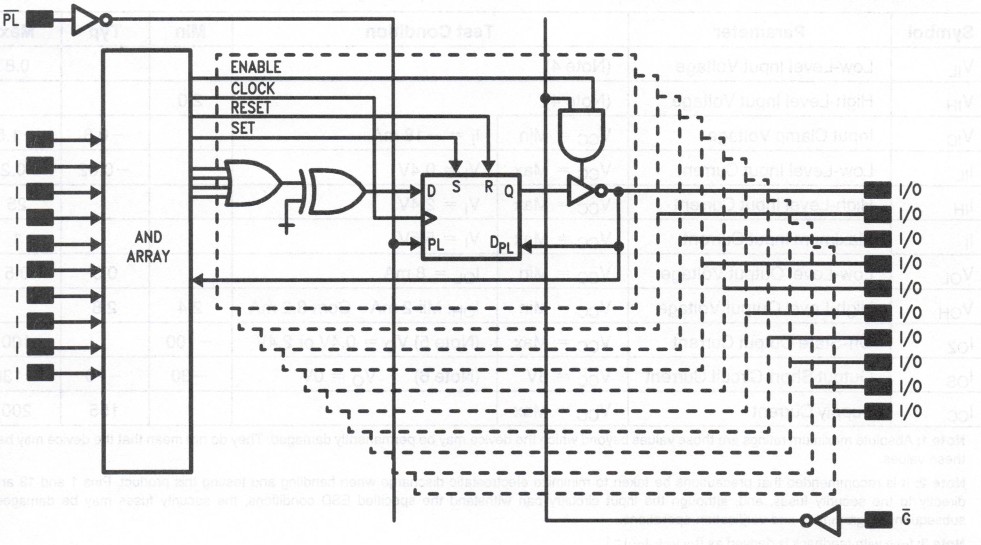
The PAL20RA10 is made up of ten Output Logic Macro Cells (OLMC). Four AND array outputs feed into the fixed OR-gate for each OLMC to generate the device's output functions. Four other AND array outputs are used for control functions in the OLMC. With a robust mixture of logic derived controlled functions and selectable output data paths, the PAL20RA10 provides an ideal solution for registered random logic applications.

This device is housed in a 24-pin 300 mil DIP. A 28-pin PCC package is also available. It can be programmed by most PAL programmers.

Features

- Programmable asynchronous set and reset
- Individually programmable clocks
- Programmable and hard-wired TRI-STATE® outputs
- Programmable output polarity
- Registers can be bypassed individually
- Register preload guarantees testability
- Outputs can be reconfigured as inputs
- Power-up reset for registered outputs
- Fully supported by National PLAN™ development software
- A variety of JEDEC-compatible programming equipment and design development software available
- Security fuse prevents direct copying of logic patterns

Block Diagram—PAL20RA10



TL/L/8702-24

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Operating	Programming
Supply Voltage V_{CC}	7.0V	12.0V
Input Voltage	5.5V	22.0V

	Operating	Programming
Off-State Output Voltage	5.5V	12.0V
Storage Temperature		−65°C to +150°C
ESD Tolerance (Note 2)		1000V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Military Min	Military Nom	Military Max	Commercial Min	Commercial Nom	Commercial Max	Units
V_{CC}	Supply Voltage		4.5	5	5.5	4.75	5	5.25	V
T_A	Operating Free-Air Temperature		−55			0		75	°C
T_C	Operating Case Temperature				125				°C
t_W	Pulse Width of Clocking Input (High/Low)		25	13		20	13		ns
t_{WP}	Pulse Width of Preload ($\overline{PL}$) Input (Low)		45	15		35	15		ns
t_{SU}	Setup Time from Input or Feedback to Clocking Input		25	10		20	10		ns
t_{SUP}	Setup Time from Input to $\overline{PL}$ High		30	5		25	5		ns
t_H	Hold Time of Input after Clocking Input	Polarity Fuse Intact	10	−2		10	−2		ns
		Polarity Fuse Blown	0	−6		0	−6		
t_{HP}	Hold Time of Input after $\overline{PL}$ High		30	5		25	5		ns
f_{CLK}	Clock Frequency (Note 3)	With Feedback		37	16.6		37	20	MHz
		Without Feedback		38.5	20		38.5	25	MHz

Electrical Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Condition		Min	Typ	Max	Units
V_{IL}	Low-Level Input Voltage	(Note 4)				0.8	V
V_{IH}	High-Level Input Voltage	(Note 4)		2.0			V
V_{IC}	Input Clamp Voltage	V_{CC} = Min	I_I = −18 mA		−0.8	−1.5	V
I_{IL}	Low-Level Input Current	V_{CC} = Max	V_I = 0.4V		−0.02	−0.25	mA
I_{IH}	High-Level Input Current	V_{CC} = Max	V_I = 2.4V			25	μA
I_I	Maximum Input Current	V_{CC} = Max	V_I = 5.5V			1	mA
V_{OL}	Low-Level Output Voltage	V_{CC} = Min	I_{OL} = 8 mA		0.3	0.5	V
V_{OH}	High-Level Output Voltage	V_{CC} = Min	I_{OH}: Mil-2 mA Com-3.2 mA	2.4	2.8		V
I_{OZ}	Off-State Output Current	V_{CC} = Max	(Note 5) V_O = 0.4V or 2.4V	−100		100	μA
I_{OS}	Output Short-Circuit Current	V_{CC} = 5V	(Note 6) V_O = 0V	−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max			155	200	mA

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. They do not mean that the device may be operated at these values.

Note 2: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. Pins 1 and 13 are connected directly to the security fuses, and, although the input circuitry can withstand the specified ESD conditions, the security fuses may be damaged preventing subsequent programming and verification operations.

Note 3: f_{CLK} with feedback is derived as $(t_{CLK} + t_{SU})^{-1}$.
f_{CLK} without feedback is derived as $(2t_W)^{-1}$.

Note 4: These are absolute voltages with respect to the ground pin on the device and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.

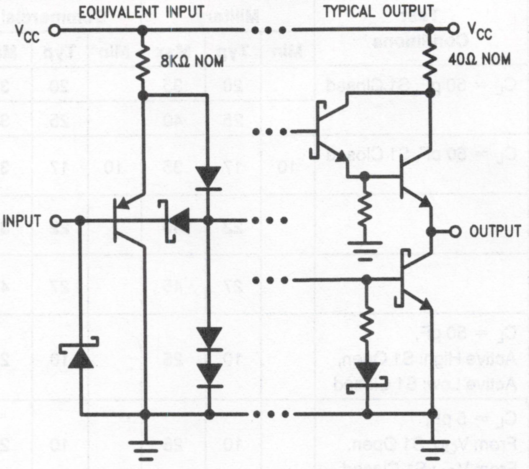
Note 5: I/O leakage as the worst case of I_{OZX} or I_{IX}, e.g. I_{IL} and I_{OZL}.

Note 6: During I_{OS} measurement, only one output at a time should be grounded. Permanent damage otherwise may result.

Switching Characteristics Over Recommended Operating Conditions

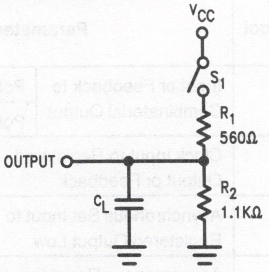
Symbol	Parameter		Test Conditions	Military			Commercial			Units
				Min	Typ	Max	Min	Typ	Max	
t_{PD}	Input or Feedback to Combinatorial Output	Polarity Fuse Intact	C_L = 50 pF, S1 Closed		20	35		20	30	ns
		Polarity Fuse Blown			25	40		25	35	
t_{CLK}	Clock Input to Registered Output or Feedback		C_L = 50 pF, S1 Closed	10	17	35	10	17	30	ns
t_S	Asynchronous Set Input to Registered Output Low				22	40		22	35	ns
t_R	Asynchronous Reset Input to Registered Output High				27	45		27	40	ns
t_{PZXG}	$\overline{G}$ Pin to Output Enabled		C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		10	25		10	20	ns
t_{PXZG}	$\overline{G}$ Pin to Output Disabled		C_L = 5 pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		10	25		10	20	ns
t_{PZXI}	Input to Output Enabled via Product Term		C_L = 50 pF, Active High: S1 Open, Active Low: S1 Closed		18	35		18	30	ns
t_{PXZI}	Input to Output Disabled via Product Term		C_L = 5 pF, From V_{OH}: S1 Open, From V_{OL}: S1 Closed		15	35		15	30	ns
t_{RESET}	Power-Up to Registered Output High				600	1000		600	1000	ns

Schematic of Inputs and Outputs


EQUIVALENT INPUT
TYPICAL OUTPUT
V_{CC}
V_{CC}
8KΩ NOM
40Ω NOM
INPUT
OUTPUT

TL/L/8702-3

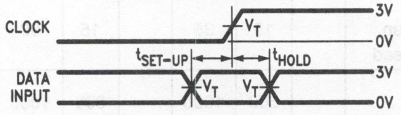
Test Load


V_{CC}
S_1
R_1
560Ω
OUTPUT
C_L
R_2
1.1KΩ

TL/L/8702-4

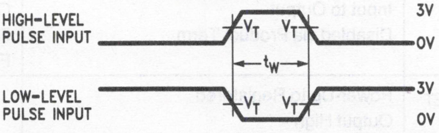
Test Waveforms

Set-Up and Hold


CLOCK
DATA INPUT
V_T
V_T
V_T
3V
0V
3V
0V
t_{SET-UP}
t_{HOLD}

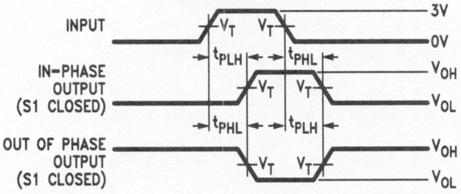
TL/L/8702-25

Pulse Width


HIGH-LEVEL PULSE INPUT
LOW-LEVEL PULSE INPUT
V_T
V_T
V_T
V_T
t_W
3V
0V
3V
0V

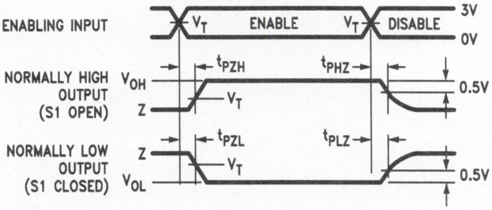
TL/L/8702-27

Propagation Delay


INPUT
IN-PHASE OUTPUT (S1 CLOSED)
OUT OF PHASE OUTPUT (S1 CLOSED)
V_T
V_T
t_{PLH}
t_{PHL}
t_{PHL}
t_{PLH}
3V
0V
V_{OH}
V_{OL}
V_{OH}
V_{OL}

TL/L/8702-26

Enable and Disable


ENABLING INPUT
V_T
ENABLE
V_T
DISABLE
3V
0V
NORMALLY HIGH OUTPUT (S1 OPEN)
V_{OH}
Z
t_{PZH}
t_{PHZ}
V_T
0.5V
NORMALLY LOW OUTPUT (S1 CLOSED)
Z
V_{OL}
t_{PZL}
t_{PLZ}
V_T
0.5V

TL/L/8702-28

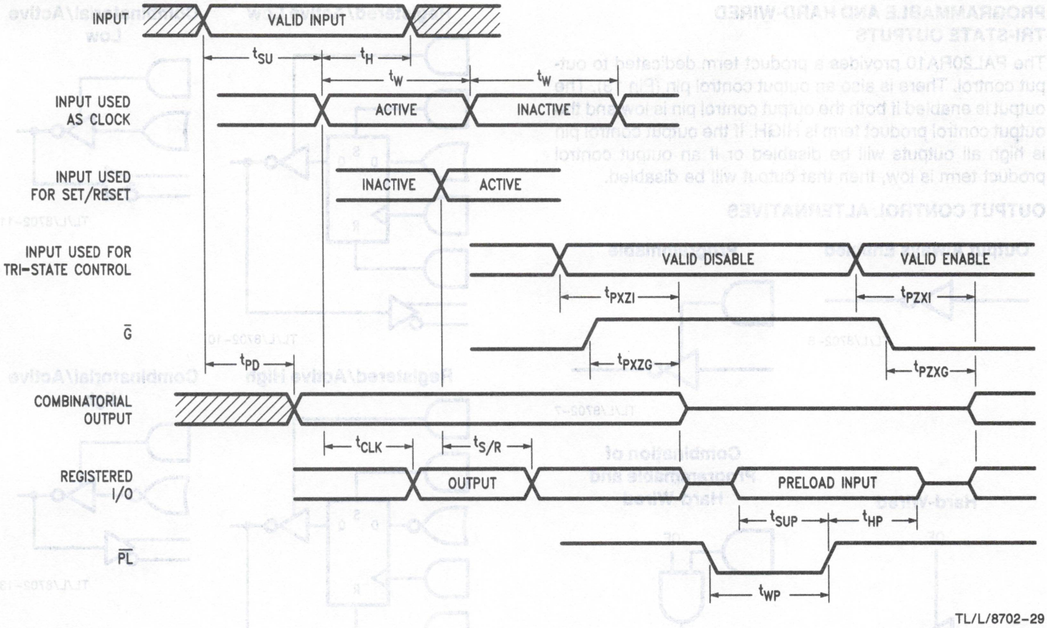
Notes:

$V_T = 1.5V$

C_L includes probe and jig capacitance.

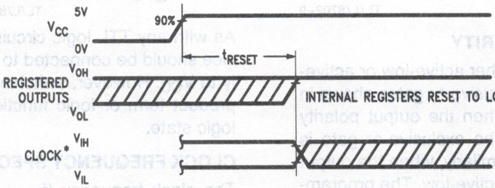
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUT
VALID INPUT
tSU
tH
tW
tW
INPUT USED AS CLOCK
ACTIVE
INACTIVE
INPUT USED FOR SET/RESET
INACTIVE
ACTIVE
INPUT USED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
tPXZI
tPZXI
G̅
tPXZG
tPZXG
tPD
COMBINATORIAL OUTPUT
tCLK
tS/R
REGISTERED I/O
OUTPUT
PRELOAD INPUT
tSUP
tHP
PL̅
tWP

TL/L/8702-29

Power-Up Reset Waveform


5V
VCC
0V
90%
tRESET
VOH
REGISTERED OUTPUTS
VOL
INTERNAL REGISTERS RESET TO LOGIC 0
VIH
CLOCK*
VIL

TL/L/8702-30

*The clock input should not be switched from low to high until after time t_{RESET}.

Functional Description

The PAL20RA10 logic array consists of 20 complementary input lines and 80 product-term lines with a programmable fuse link at each intersection (3200 fuses). The product terms are organized into ten groups of eight each. Four of the eight product terms in each group connect into an OR-gate to produce the sum-of-products logic function. The remaining four product terms in each group are used for control functions in the Output Logic Macro Cell (OLMC) as shown in *Figure 1*.

An unprogrammed (intact) fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term; programming the fuse removes the connection. A product term is satisfied (logically true) while all of the input lines connected to it (via unprogrammed fuses) are in the high logic state. Therefore, if both the true and complement of at least one array input is left connected to a product line, that product term is always held in the low logic state (which is the state of all product terms in an unprogrammed device). Conversely, if all fuses on a product term were programmed, the product term and the resulting logic function would be held in the high state.

PROGRAMMABLE SET AND RESET

In each cell, two product lines are dedicated to asynchronous set and reset. If the set product line is high, the register output becomes a logic 1, the output pin becomes a 0. If the reset product line is high, the register output becomes a logic 0, the output pin becomes a 1. The operation of the programmable set and reset overrides the clock.

INDIVIDUALLY PROGRAMMABLE REGISTER BYPASS

If both the set and reset product lines are high, the sum-of-products bypasses the register and appears immediately at the output, thus making the output combinatorial. This allows each output to be configured in the registered or combinatorial mode.

PROGRAMMABLE CLOCK

One of the product lines in each group is connected to the clock. This provides the user with the additional flexibility of a programmable clock, so each output can be clocked independently of all the others.

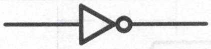
Functional Description (Continued)

PROGRAMMABLE AND HARD-WIRED TRI-STATE OUTPUTS

The PAL20RA10 provides a product term dedicated to output control. There is also an output control pin (Pin 13). The output is enabled if both the output control pin is low and the output control product term is HIGH. If the output control pin is high all outputs will be disabled or if an output control product term is low, then that output will be disabled.

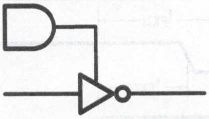
OUTPUT CONTROL ALTERNATIVES

Output Always Enabled



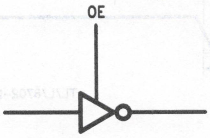
TL/L/8702-6

Programmable



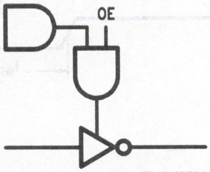
TL/L/8702-7

Hard-Wired



TL/L/8702-8

Combination of Programmable and Hard-Wired



TL/L/8702-9

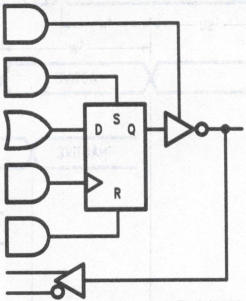
PROGRAMMABLE OUTPUT POLARITY

The outputs can be programmed either active-low or active-high. This is represented by the exclusive-or gates shown in the PAL20RA10 Logic Diagram. When the output polarity fuse is blown, the lower input to the exclusive-or gate is high, so the output is active-high. Similarly, when the output polarity fuse is intact, the output is active-low. The programmable output polarity features allows the user a higher degree of flexibility when writing equations.

POWER-UP RESET

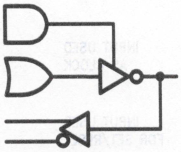
The PAL20RA10 device resets all registers to a low state upon power-up (active-low outputs assume high logic levels if enabled). This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

Registered/Active Low



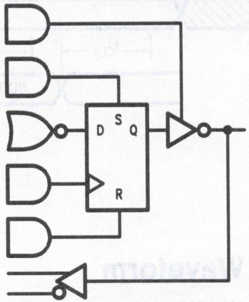
TL/L/8702-10

Combinatorial/Active Low



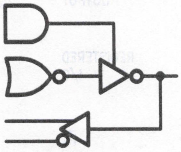
TL/L/8702-11

Registered/Active High



TL/L/8702-12

Combinatorial/Active High



TL/L/8702-13

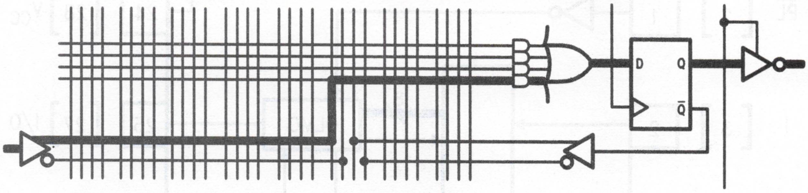
As with any TTL logic circuits, unused inputs to a PAL device should be connected to ground, V_{OL}, V_{OH}, or resistively to V_{CC}. However, switching any input not connected to a product term or logic function has no effect on its output logic state.

CLOCK FREQUENCY SPECIFICATION

The clock frequency (f_{CLK}) parameter specifies the maximum speed at which a registered PAL device is guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, where the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e., based only on external inputs), the minimum required cycle period ($f_{CLK}{}^{-1}$ without feedback) is defined as the greater of the minimum clock period (t_W high + t_W low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period ($f_{CLK}{}^{-1}$ with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set-up on the inputs to the registers before the end of each cycle.

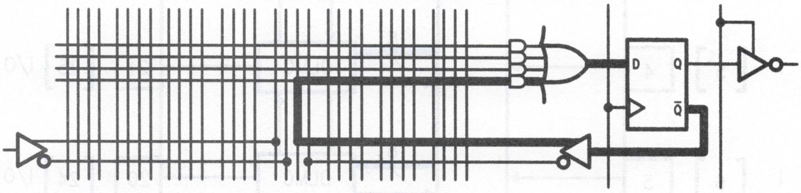
Functional Description (Continued)

Typical Registered Logic Function Without Feedback

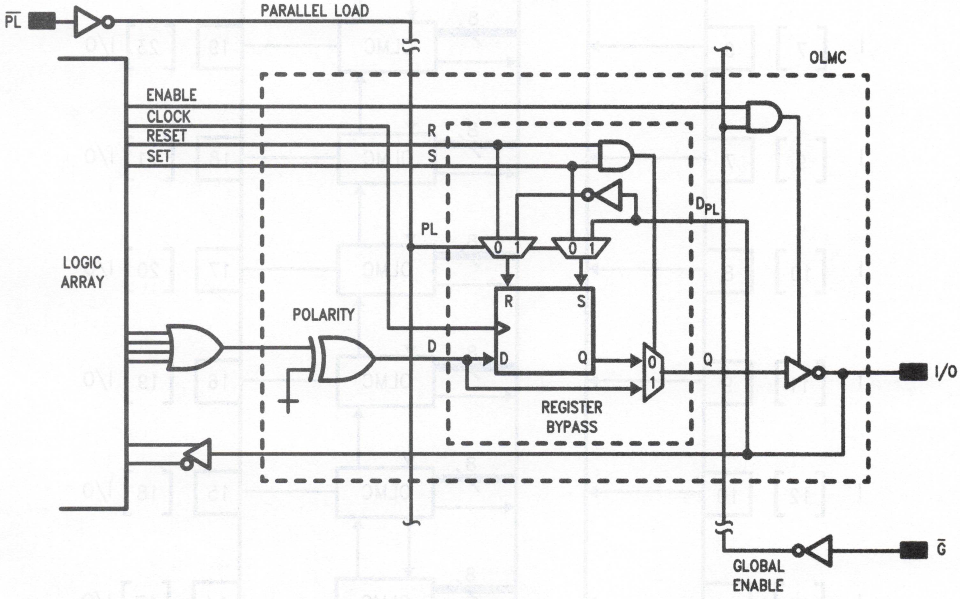


TL/L/8702-32

Typical Registered Logic Function With Feedback



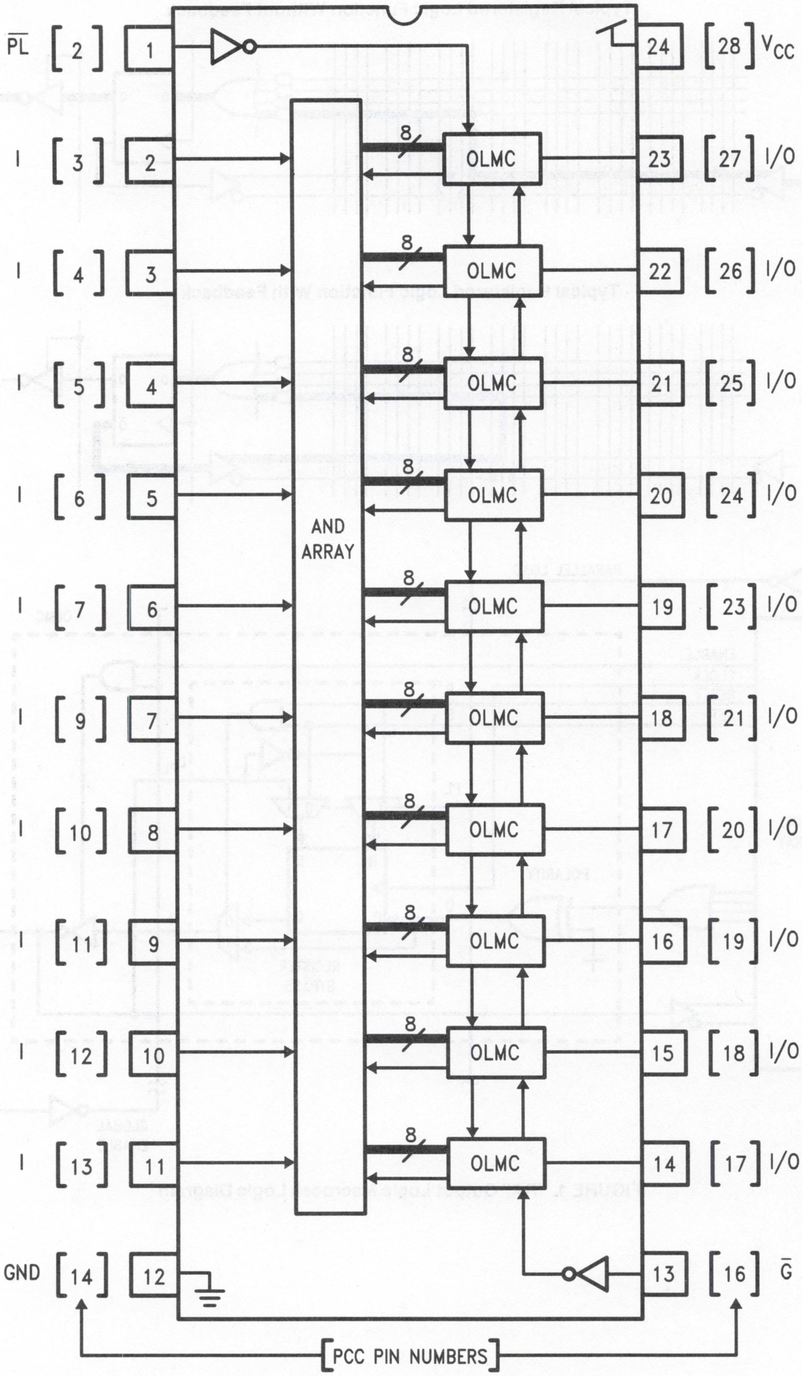
TL/L/8702-33


P̄L̄
PARALLEL LOAD
OLMC
ENABLE
CLOCK
RESET
SET
R
S
D_PL
PL
0 1
0 1
LOGIC ARRAY
R
S
POLARITY
D
D
Q
0
1
Q
I/O
REGISTER BYPASS
Ḡ
GLOBAL ENABLE

TL/L/8702-34

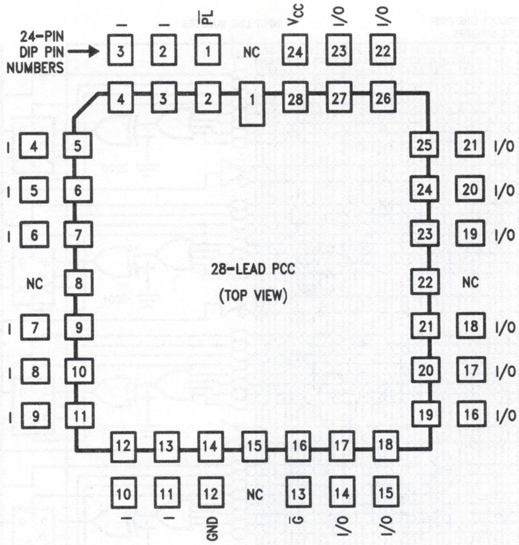
FIGURE 1. "RA" Output Logic Macrocell Logic Diagram

24-Pin PAL20RA10 Block Diagram—DIP Connections


PL
[2]
1
I
[3]
2
I
[4]
3
I
[5]
4
I
[6]
5
I
[7]
6
I
[9]
7
I
[10]
8
I
[11]
9
I
[12]
10
I
[13]
11
GND
[14]
12
AND
ARRAY
8
OLMC
24
[28]
VCC
23
[27]
I/O
22
[26]
I/O
21
[25]
I/O
20
[24]
I/O
19
[23]
I/O
18
[21]
I/O
17
[20]
I/O
16
[19]
I/O
15
[18]
I/O
14
[17]
I/O
13
[16]
G
[PCC PIN NUMBERS]

TL/L/8702–36

28-Lead PCC Connection Diagram


24-PIN DIP PIN NUMBERS
I 3 4
I 2 3
PL 1 2
NC 1
VCC 24 28
I/O 23 27
I/O 22 26
I 4 5
I 5 6
I 6 7
NC 8
I 7 9
I 8 10
I 9 11
25 21 I/O
24 20 I/O
23 19 I/O
22 NC
21 18 I/O
20 17 I/O
19 16 I/O
I 10 12
I 11 13
GND 12 14
NC 15
G 13 16
I/O 14 17
I/O 15 18
28-LEAD PCC
(TOP VIEW)

TL/L/8702-31

Note: For availability of old (Non-JEDEC) pinout, please contact your local National Semiconductor sales representative or distributor.

Output Register Preload

Register preload allows any arbitrary state to be loaded into the PAL output registers. This allows complete logic verification, including states that are impossible or impractical to reach. To use the preload feature, first disable the outputs by bringing $\overline{OE}$ high, and present the data at the output pins. A low-level on the preload pin ($\overline{PL}$) will then load the data into the registers. Note that voltage levels applied to the $\overline{PL}$ inputs are standard TTL levels.

Security Fuse

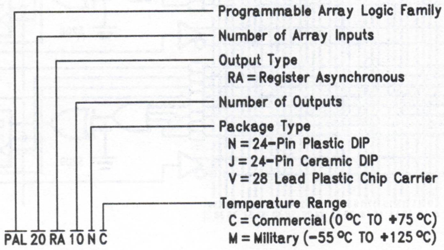
Security fuses are provided on all National PAL devices which, when programmed, inhibit any further programming or verifying operations. This feature prevents direct copying of proprietary logic patterns. The security fuses should be programmed only after programming and verifying all other device fuses. Register preload is not affected by the security fuses.

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a variety of programming equipment. Many software packages and programming units support a wide variety of programmable logic products as well. The PLAN software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

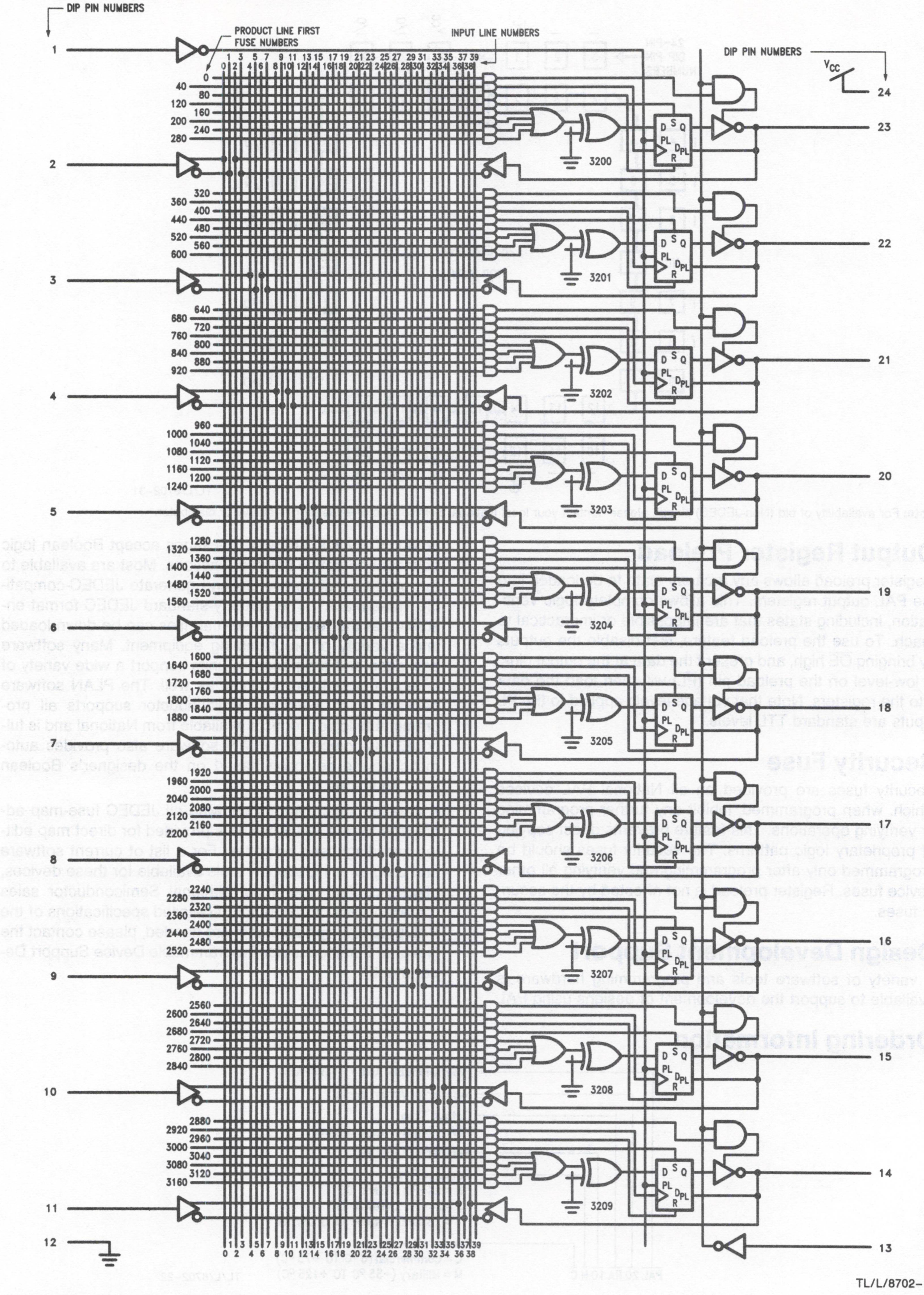
A detailed logic diagram showing all JEDEC fuse-map addresses for the PAL20RA10 is provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

Ordering Information


Programmable Array Logic Family
Number of Array Inputs
Output Type
RA = Register Asynchronous
Number of Outputs
Package Type
N = 24-Pin Plastic DIP
J = 24-Pin Ceramic DIP
V = 28 Lead Plastic Chip Carrier
Temperature Range
C = Commercial (0 °C TO +75 °C)
M = Military (−55 °C TO +125 °C)
PAL 20 RA 10 N C

TL/L/8702-22

Logic Diagram—PAL20RA10


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
0 40 80 120 160 200 240 280
320 360 400 440 480 520 560 600
640 680 720 760 800 840 880 920
960 1000 1040 1080 1120 1160 1200 1240
1280 1320 1360 1400 1440 1480 1520 1560
1600 1640 1680 1720 1760 1800 1840 1880
1920 1960 2000 2040 2080 2120 2160 2200
2240 2280 2320 2360 2400 2440 2480 2520
2560 2600 2640 2680 2720 2760 2800 2840
2880 2920 2960 3000 3040 3080 3120 3160
1 2 3 4 5 6 7 8 9 10 11 12
24 23 22 21 20 19 18 17 16 15 14 13
3200 3201 3202 3203 3204 3205 3206 3207 3208 3209
D S Q
PL
D_{PL}
R

TL/L/8702-15

JEDEC Logic Array Fuse Number = Product Line First Fuse Number + Input Line Number.

National Semiconductor

GAL16V8 Generic Array Logic

General Description

The NSC E2CMOS™ GAL® device combines a high performance CMOS process with electrically erasable floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels.

The 20-pin GAL16V8 features 8 programmable Output Logic Macrocells (OLMCs) allowing each TRI-STATE® output to be configured by the user. Additionally, the GAL16V8 is capable of emulating, in a functional/fuse map/parametric compatible device, all common 20-pin PAL® device architectures.

Programming is accomplished using readily available hardware and software tools. NSC guarantees a minimum 100 erase/write cycles.

Unique test circuitry and reprogrammable cells allow complete AC, DC, cell and functionality testing during manufacture. Therefore, NSC guarantees 100% field programmability and functionality of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.

Features

- Electrically erasable cell technology
  - — Reconfigurable logic
  - — Reprogrammable cells
  - — Guaranteed 100% yields
- High performance E2CMOS technology
  - — Low power: 45 mA/90 mA max active
  - — High Speed: 20 ns–35 ns max access
- Eight output logic macrocells
  - — Maximum flexibility for complex logic designs
  - — Also emulates 20-pin PAL devices with full function/fuse map/parametric compatibility
- Preload and power-on reset of all registers
  - — 100% functional testability
- Fully supported by National PLAN™ development software
- High speed programming algorithm
- Security cell prevents copying logic

PAL Replacement by Device Type

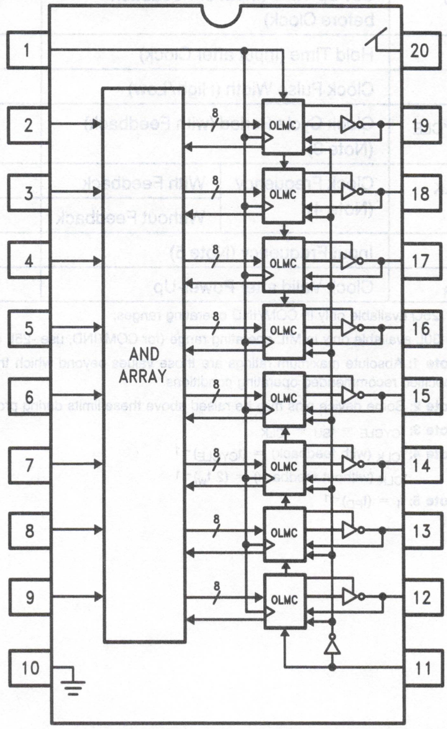
"Small-PAL" Mode				"Registered-PAL" Mode			"Medium-PAL" Mode
10L8	12L6	14L4	16L2	16R8	16R6	16R4	16L8
10H8	12H6	14H4	16H2	16RP8	16RP6	16RP4	16H8
10P8	12P6	14P4	16P2				16P8

PAL Replacement by Speed/Power

PAL			GAL
Rev	Speed	Power*	Rev
B-2	25 ns	90 mA	25L or 20L or 25Q (45 mA)
A	25 ns	180 mA	25L or 20L (90 mA)
B-2 MIL	30 ns	90 mA	30L or 25L (110 mA) or 30Q (55 mA)
A MIL	30 ns	180 mA	30L or 25L (110 mA)
B-4	35 ns	45 mA	30Q
A-2	35 ns	90 mA	30Q (45 mA)
STD	35 ns	180 mA	25L (90 mA)

*Shown for Medium PAL products

Block Diagram—GAL16V8



TL/L/9344-1

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−2.5V to V_{CC} +1.0V
Off-State Output Voltage (Note 2)	−2.5V to V_{CC} +1.0V
Output Current	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	260°C
ESD Tolerance	500V

C_{ZAP} = 100 pF
R_{ZAP} = 150Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

SUPPLY VOLTAGE AND TEMPERATURE

Symbol	Parameter	Commercial			Industrial			Military			Units
		Min	Nom	Max	Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	4.5	5	5.5	4.5	5	5.5	V
T_A	Operating Free-Air Temperature	0	25	75	−40	25	85	−55	25		°C
T_C	Operating Case Temperature									125	°C

AC TIMING REQUIREMENTS

Symbol	Parameter	GAL16V8-20L COM/IND		GAL16V8-25Q GAL16V8-25L COM/IND MIL*		GAL16V8-30Q GAL16V8-30L COM/IND* MIL		Units
		Min	Max	Min	Max	Min	Max	
t_{SU}	Set-Up Time (Input or Feedback before Clock)	15		20		25		ns
t_H	Hold Time (Input after Clock)	0		0		0		ns
t_W	Clock Pulse Width (High/Low)	12		15		15		ns
t_{CYCLE}	Clock Cycle Period (with Feedback) (Note 3)	30		35		45		ns
f_{CLK}	Clock Frequency (Note 4) With Feedback		33.3		28.5		22.2	MHz
	Clock Frequency (Note 4) Without Feedback		41.6		33.3		33.3	
f_I	Input Frequency (Note 5)		50.0		40.0		33.3	
t_{PR}	Clock Valid after Power-Up		100		100		100	ns

* -25Q available only in COM/IND operating ranges;
-30L available only in MIL operating range (for COM/IND, use -25L or -30Q).

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: $t_{CYCLE} = t_{SU} + t_{CLK}$

Note 4: f_{CLK} (with feedback) = $(t_{CYCLE})^{-1}$
f_{CLK} (without feedback) = $(2\ t_W)^{-1}$

Note 5: $f_I = (t_{PD})^{-1}$

Electrical Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Conditions	Temperature Range	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage			2.0		V_{CC} + 1	V
V_{IL}	Low Level Input Voltage			−1.0		0.8	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = −3.2 mA	COM/IND	2.4			V
		I_{OH} = −2.0 mA	MIL	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = 24 mA	COM/IND			0.5	V
		I_{OL} = 12 mA	MIL			0.5	V
I_{OZH}	High Level Off State Output Current	V_{CC} = Max, V_O = V_{CC} (Max)				10	µA
I_{OZL}	Low Level Off State Output Current	V_{CC} = Max, V_O = GND				−10	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = V_{CC} (Max)				10	µA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = V_{CC} (Max)				10	µA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = GND				−10	µA
I_{OS}*	Output Short Circuit Current	V_{CC} = 5.0V, V_O = GND		−30		−130	mA
I_{CC}	Supply Current: Quarter Power (GAL16V8-25Q, -30Q)	f = 15 MHz, V_{CC} = Max	COM			45	mA
			MIL/IND			55	mA
	Supply Current: Half Power (GAL16V8-20L, -25L, -30L)	f = 15 MHz, V_{CC} = Max	COM			90	mA
			MIL/IND			110	mA
C_I	Input Capacitance	V_{CC} = 5.0V, V_I = 2.0V				12	pF
$C_{I/O}$	I/O Capacitance	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V				15	pF

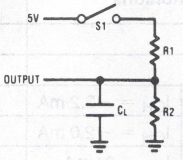
*One output at a time for a maximum duration of one second @ 25°C.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Conditions	GAL16V8-20L		GAL16V8-25Q GAL16V8-25L		GAL16V8-30Q GAL16V8-30L		Units
			COM/IND		COM/IND MIL*		COM/IND* MIL		
			Min	Max	Min	Max	Min	Max	
t_{PD}	Input or Feedback to Combinatorial Output	S1 Closed, C_L = 50 pF		20		25		30	ns
t_{CLK}	Clock to Registered Output or Feedback	S1 Closed, C_L = 50 pF		15		15		20	ns
t_{PZXG}	$\overline{G}$ ↓ to Registered Output Enabled	Active High: S1 Open, C_L = 50 pF Active Low: S1 Closed, C_L = 50 pF		18		20		25	ns
t_{PXZG}	$\overline{G}$ ↑ to Registered Output Disabled	From V_{OH}: S1 Open, C_L = 5 pF From V_{OL}: S1 Closed, C_L = 5 pF		18		20		25	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	Active High: S1 Open, C_L = 50 pF Active Low: S1 Closed, C_L = 50 pF		20		25		30	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	From V_{OH}: S1 Open, C_L = 5 pF From V_{OL}: S1 Closed, C_L = 5 pF		20		25		30	ns
t_{RESET}	Power-Up to Registered Output High	S1 Closed, C_L = 50 pF		45		45		45	µs

* -25Q available only in COM/IND operating ranges;
-30L available only in MIL operating range (for COM/IND, use -25L or -30Q).

AC Test Load

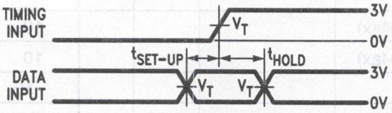

5V
S1
R1
OUTPUT
CL
R2

MIL
R1 = 390
R2 = 750
COM/IND
R1 = 200
R2 = 390

TL/L/9344-21

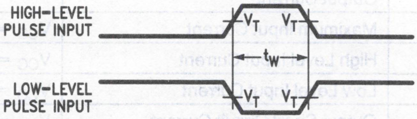
Test Waveforms

Setup and Hold


TIMING INPUT
VT
3V
0V
tSET-UP
tHOLD
DATA INPUT
VT
VT
3V
0V

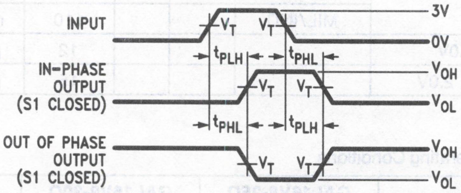
TL/L/9344-23

Pulse Width


HIGH-LEVEL PULSE INPUT
VT
VT
tW
LOW-LEVEL PULSE INPUT
VT
VT

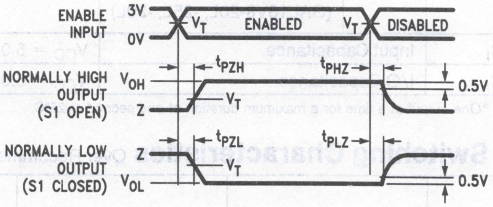
TL/L/9344-24

Propagation Delay


INPUT
VT
VT
3V
0V
tPLH
tPHL
IN-PHASE OUTPUT (S1 CLOSED)
VT
VT
VOH
VOL
tPHL
tPLH
OUT OF PHASE OUTPUT (S1 CLOSED)
VT
VT
VOH
VOL

TL/L/9344-25

Enable and Disable


ENABLE INPUT
3V
0V
VT
ENABLED
VT
DISABLED
tPZH
tPHZ
NORMALLY HIGH OUTPUT (S1 OPEN)
VOH
Z
VT
0.5V
tPZL
tPLZ
NORMALLY LOW OUTPUT (S1 CLOSED)
VOL
VT
0.5V

TL/L/9344-26

Notes:

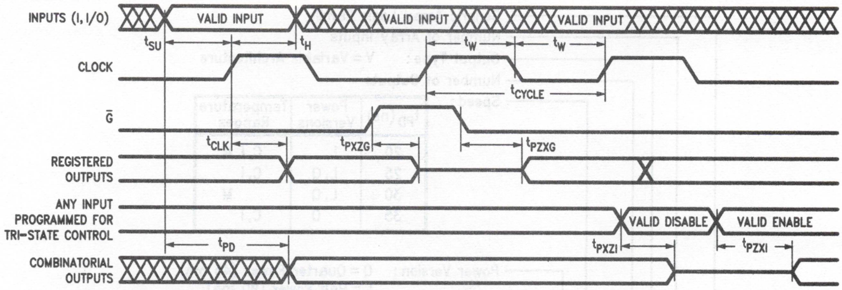
C_L includes probe and jig capacitance.

V_T = 1.5V.

Test inputs have rise and fall times of 5 ns between 0.3V and 2.7V.

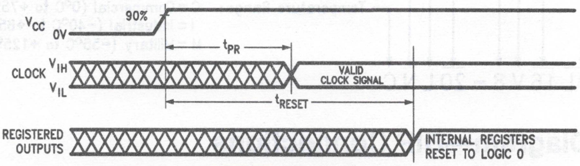
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS (I, I/O)
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
t_{CYCLE}
$\bar{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/9344-7

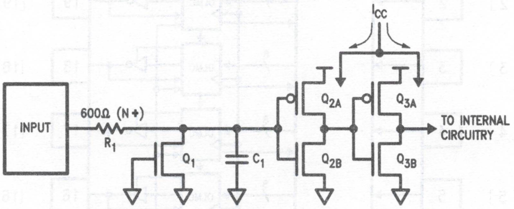
Power-Up Reset Waveforms


90%
V_{CC}
0V
t_{PR}
CLOCK
V_{IH}
V_{IL}
VALID CLOCK SIGNAL
t_{RESET}
REGISTERED OUTPUTS
INTERNAL REGISTERS RESET TO LOGIC 0

TL/L/9344-18

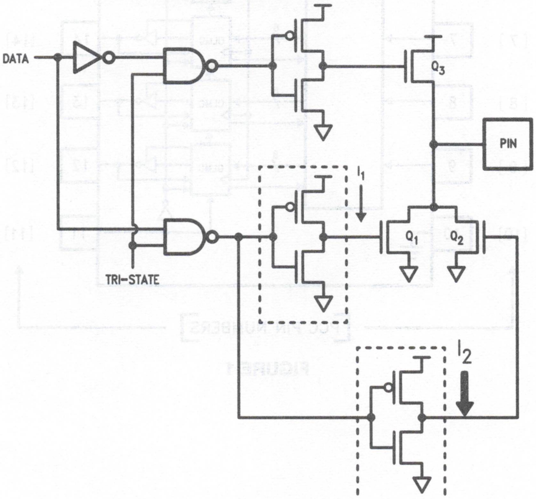
Input/Output Schematics

Input Translator/Buffer


I_{CC}
INPUT
600Ω (N+)
R_1
Q_1
C_1
Q_{2A}
Q_{2B}
Q_{3A}
Q_{3B}
TO INTERNAL CIRCUITRY

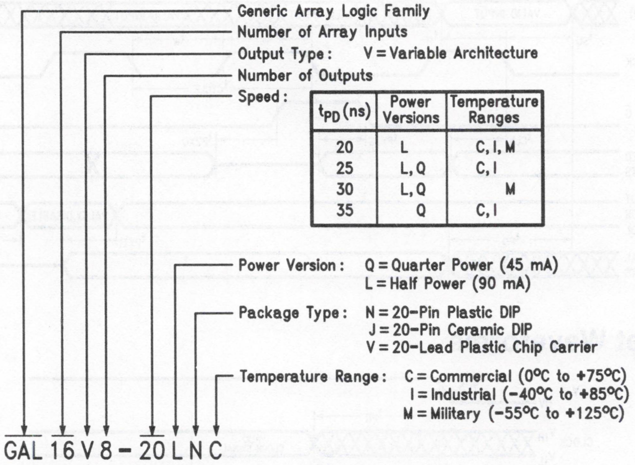
TL/L/9344-27

Phased Output Turn-On Circuit


DATA
Q_3
PIN
I_1
Q_1
Q_2
TRI-STATE
I_2

TL/L/9344-28

Ordering Information



Generic Array Logic Family

Number of Array Inputs

Output Type: V = Variable Architecture

Number of Outputs

Speed:

t_{PD} (ns)	Power Versions	Temperature Ranges
20	L	C, I, M
25	L, Q	C, I
30	L, Q	M
35	Q	C, I

Power Version: Q = Quarter Power (45 mA)
L = Half Power (90 mA)

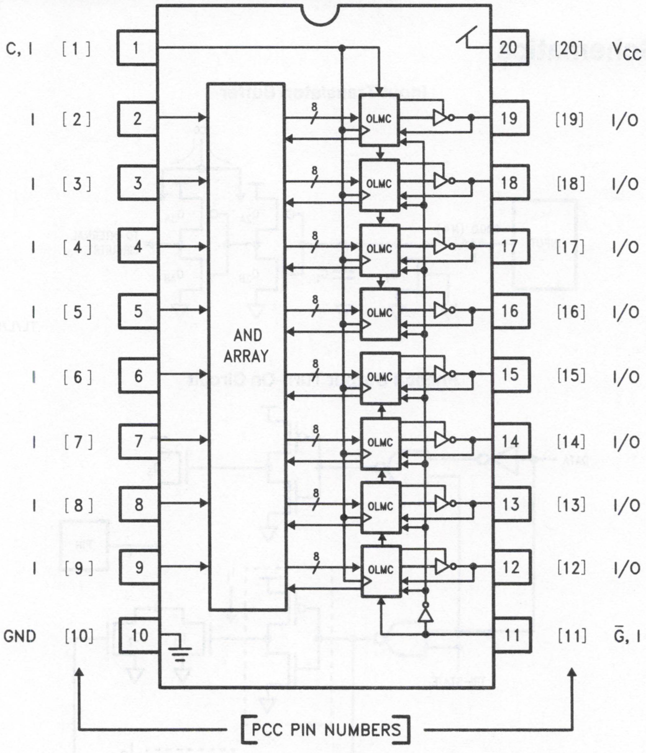
Package Type: N = 20-Pin Plastic DIP
J = 20-Pin Ceramic DIP
V = 20-Lead Plastic Chip Carrier

Temperature Range: C = Commercial (0°C to +75°C)
I = Industrial (−40°C to +85°C)
M = Military (−55°C to +125°C)

GAL 16 V 8 − 20 L N C

TL/L/9344–29

GAL16V8 Block Diagram—DIP Connections


C, I [1] 1
I [2] 2
I [3] 3
I [4] 4
I [5] 5
I [6] 6
I [7] 7
I [8] 8
I [9] 9
GND [10] 10
20 [20] V_{CC}
19 [19] I/O
18 [18] I/O
17 [17] I/O
16 [16] I/O
15 [15] I/O
14 [14] I/O
13 [13] I/O
12 [12] I/O
11 [11] $\bar{G}$, I
AND ARRAY
OLMC
8
PCC PIN NUMBERS

TL/L/9344–30

FIGURE 1

Functional Description

The GAL logic array consists of a programmable AND array with fixed OR-gate connections, similar to the bipolar PAL architecture. The logic array is organized as 16 complementary input lines crossing 64 "product term" lines with a programmable E^2PROM cell at each intersection (2048 cells). Each programmable cell may establish a connection between an input line (true or complement phase of an array input signal) and a product term. A product term is satisfied (logically true) while all of the input lines "connected" to it are in the high logic state.

The 64 product terms are organized into eight output groups with eight terms each. Seven or eight of the product terms in each output group feed into an OR-gate to produce each output logic function; one of the product terms may instead be used to control the associated TRI-STATE device output. The fundamental transfer function of each GAL output is the familiar Boolean sum-of-products. Design development software is available which accepts Boolean equations and converts them automatically into GAL programming patterns.

As shown in the GAL16V8 Block Diagram *(Figure 1)*, a total of eight output logic functions are available. Each of the AND/OR logic functions feeds into an "output logic macrocell" (OLMC). The eight OLMCs control the flow of input and output signals between the logic array and the device's I/O pins.

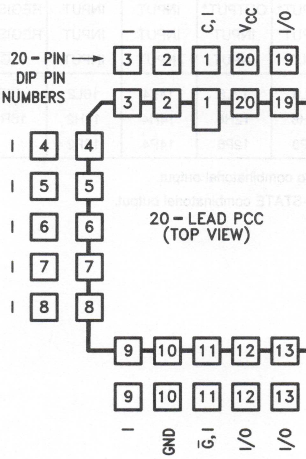
Under control of an OLMC, each output may be designated either registered or combinatorial (non-registered). In the registered output configuration, the logic function output passes through a D-type flip-flop triggered by the rising edge of the clock input. Additionally, the logic function's output polarity may be designated active-low or active-high (adjusted before the register, if present). OLMC options such as these are selected using a set of programmable architecture control cells. These architecture cells are normally configured automatically by the development software or programming hardware.

All of the possible I/O configurations of the GAL16V8 are classified into three basic modes: "Small-PAL" mode, "Registered-PAL" mode and "Medium-PAL" mode. These modes correspond to the architectures of the PAL families which the GAL16V8 can emulate. The modes determine the mixture of OLMC configurations which can be selected for the device. The OLMC Selection table (Table I) lists which functions can be selected on the device pin* 1 and pins* 11 through 19 for each of the three modes. The logic diagrams in *Figure 3* illustrate these OLMC functions.

"OUTPUT" represents the always-active combinatorial output configuration available in the "Small-PAL" mode. "REGISTER" is the registered output with register feedback available in the "Registered-PAL" mode. "I/O" is the combinatorial bidirectional I/O available in "Registered-PAL" and "Medium-PAL" modes. "TRI-STATE" is the TRI-STATE combinatorial output function appearing on pins* 12 and 19 in the "Medium-PAL" mode. "INPUT" in Table I denotes an OLMC used as a dedicated input only.

*Applies to both 20-pin DIP and 20-lead PCC packages for GAL16V8.

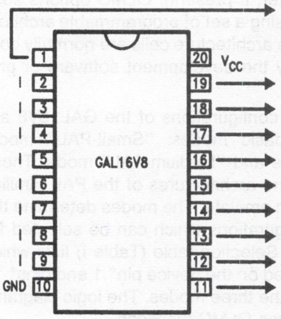
20-Lead PCC Connection Diagram



TL/L/9344-31

FIGURE 2

OLMC Selection Table



TL/L/9344-41

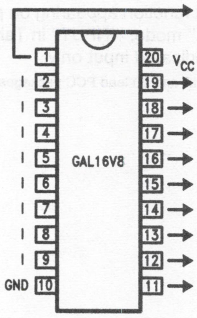
TABLE I

Pin	"Small-PAL" Mode	"Registered-PAL" Mode	"Medium-PAL" Mode
1	INPUT	CLOCK	INPUT
19	INPUT or OUTPUT*	REGISTER or I/O	TRI-STATE**
18	INPUT or OUTPUT*	REGISTER or I/O	I/O
17	INPUT or OUTPUT*	REGISTER or I/O	I/O
16	OUTPUT*	REGISTER or I/O	I/O
15	OUTPUT*	REGISTER or I/O	I/O
14	INPUT or OUTPUT*	REGISTER or I/O	I/O
13	INPUT or OUTPUT*	REGISTER or I/O	I/O
12	INPUT or OUTPUT*	REGISTER or I/O	TRI-STATE**
11	INPUT	OUPUT ENABLE ($\overline{G}$)	INPUT

*Active combinatorial output

**TRI-STATE combinatorial output

PAL Replacement Configurations



TL/L/9344-42

TABLE II

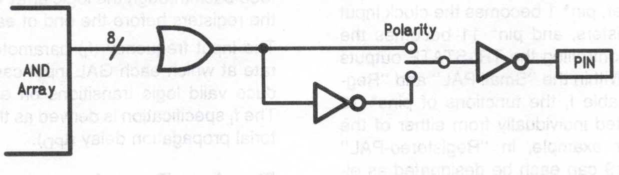
Pin	"Small PAL" Mode				"Registered-PAL"			"Medium-PAL"
1	INPUT	INPUT	INPUT	INPUT	CLOCK	CLOCK	CLOCK	INPUT
19	OUTPUT*	INPUT	INPUT	INPUT	REGISTER	I/O	I/O	TRI-STATE**
18	OUTPUT*	OUTPUT*	INPUT	INPUT	REGISTER	REGISTER	I/O	I/O
17	OUTPUT*	OUTPUT*	OUTPUT*	INPUT	REGISTER	REGISTER	REGISTER	I/O
16	OUTPUT*	OUTPUT*	OUTPUT*	OUTPUT*	REGISTER	REGISTER	REGISTER	I/O
15	OUTPUT*	OUTPUT*	OUTPUT*	OUTPUT*	REGISTER	REGISTER	REGISTER	I/O
14	OUTPUT*	OUTPUT*	OUTPUT*	INPUT	REGISTER	REGISTER	REGISTER	I/O
13	OUTPUT*	OUTPUT*	INPUT	INPUT	REGISTER	REGISTER	I/O	I/O
12	OUTPUT*	INPUT	INPUT	INPUT	REGISTER	I/O	I/O	TRI-STATE**
11	INPUT	INPUT	INPUT	INPUT	$\overline{G}$	$\overline{G}$	$\overline{G}$	INPUT
EMULATED PAL PRODUCTS	10L8 10H8 10P8	12L6 12H6 12P6	14L4 14H4 14P4	16L2 16H2 16P2	16R8 16RP8	16R6 16RP6	16R4 16RP4	16L8 16H8 16P8

*Active combinatorial output.

**TRI-STATE combinatorial output.

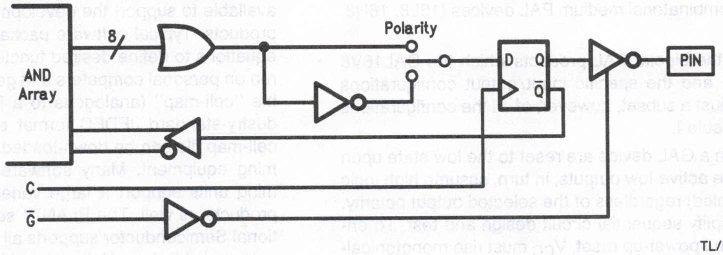
OLMC Configurations

OUTPUT (Active Combinatorial Output)


8
AND Array
Polarity
PIN

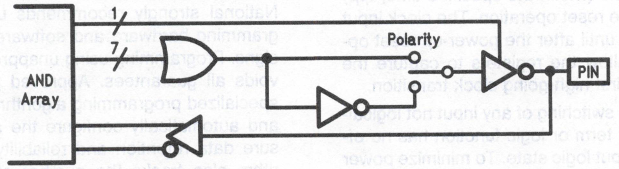
TL/L/9344-32

REGISTER (Registered Output)


8
AND Array
Polarity
D
Q
$\overline{Q}$
PIN
C
$\overline{G}$

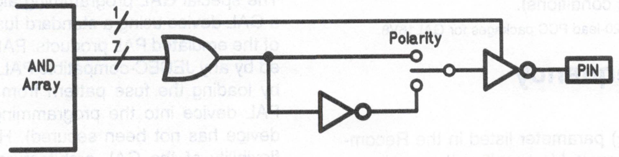
TL/L/9344-33

I/O (Combinatorial Input/Output)


1
7
AND Array
Polarity
PIN

TL/L/9344-34

TRI-STATE (TRI-STATE Combinatorial Output)


1
7
AND Array
Polarity
PIN

TL/L/9344-43

FIGURE 3

Functional Description (Continued)

In the "Small-PAL" and "Medium-PAL" modes (Table I), pins* 1 and 11 are always dedicated inputs. In the "Registered-PAL" mode, however, pin* 1 becomes the clock input controlling all OLMC registers, and pin* 11 becomes the output enable ($\overline{G}$) input controlling the TRI-STATE outputs of all registered OLMCs. Within the "Small-PAL" and "Registered-PAL" modes in Table I, the functions of pins* 12 through 19 can be selected individually from either of the two functions listed. For example, in "Registered-PAL" mode, pins* 12 through 19 can each be designated as either a registered output or a combinatorial I/O. The "Medium-PAL" mode represents a single fixed configuration used to emulate combinatorial medium PAL devices (16L8, 16H8, 16P8).

Table II lists the bipolar PAL products which the GAL16V8 can emulate, and the specific input/output configurations used. This is just a subset, however, of all the configurations provided in Table I.

All registers in a GAL device are reset to the low state upon power-up. The active-low outputs, in turn, assume high logic levels (if enabled) regardless of the selected output polarity. This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible (within the specified time, t_{PR}) to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

It should be noted that the switching of any input not logically connected to a product term or logic function has no effect on the associated output logic state. To minimize power consumption, however, unused inputs should be connected to a stable logic level such as ground or V_{CC} (CMOS GAL inputs may be tied directly to the supply voltage without causing excessive loading conditions).

*Applies to both 20-pin DIP and 20-lead PCC packages for GAL16V8.

Clock/Input Frequency Specifications

The clock frequency (f_{CLK}) parameter listed in the Recommended Operating Conditions table specifies the maximum speed at which the GAL registers are guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, when the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e. based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_w high + t_w low) and the minimum "data window" period (t_{SU} + t_H). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period ($t_{CYCLE} = f_{CLK}^{-1}$ with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set up on the inputs to the registers before the end of each cycle.

The input frequency (f_I) parameter specifies the maximum rate at which each GAL input can be toggled and still produce valid logic transitions on each combinatorial output. The f_I specification is derived as the inverse of the combinatorial propagation delay (t_{PD}).

Design Development Support

A variety of software tools and programming equipment is available to support the development of designs using GAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate a JEDEC-compatible "cell-map" (analogous to a PAL "fuse-map"). The industry-standard JEDEC format ensures that the resulting cell-map file can be down-loaded into a variety of programming equipment. Many software packages and programming units support a large variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

National strongly recommends using only approved programming hardware and software for developing GAL designs. Programming using unapproved equipment generally voids all guarantees. Approved programmers incorporate specialized programming algorithms that program the array and automatically configure the architecture cells. To ensure data retention and reliability, the programming algorithm also tracks the number of programming cycles to which each GAL device has been subjected since shipment, and stores this information automatically in the device.

The special GAL programming algorithm can also program a GAL device using a standard fuse-map developed for any of the emulated PAL products. PAL fuse-maps can be created by any JEDEC-compatible PAL development software or by loading the fuse pattern from an existing programmed PAL device into the programming unit (provided the PAL device has not been secured). However, to utilize the full flexibility of the GAL architecture, true GAL development software (such as PLAN software) is recommended.

Detailed logic diagrams showing all JEDEC cell-map addresses in the GAL logic array and OLMC are provided for direct map editing and diagnostic purposes (see "Programming Details"). For a list of current software and programming support tools available for these devices, please contact your local National sales representative or distributor. If detailed specifications of the GAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support department.

Security Cell

A security cell is provided on all GAL16V8 devices as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, the circuitry enabling array access is disabled, preventing further programming or verification of the array. The security cell can be erased only in conjunction with the array during a bulk erase cycle, so the original configuration can never be examined once this cell is programmed.

Electronic Signature

Each GAL device contains an electronic signature word consisting of 64 bits of reprogrammable memory. The electronic signature word can be programmed to contain any identification information desired by the user. Some uses include pattern identification labels, revision numbers, dates, inventory control information, etc. The data stored in the electronic signature word has no effect on the functionality of the device. The information is read out of the device using the normal program verification procedure provided by the programming equipment. The information may be accessed at any time independent of the state of the security cell. National's PLAN development software allows electronic signature data to be entered by the user and downloaded to the programming equipment.

Bulk Erase

The programming equipment automatically performs a bulk erase operation prior to each programming operation. No special erase operation need be performed by the user. Bulk erase clears the logic array, architecture cells, security cell, and electronic signature information. The GAL device is thereby reverted back to its virgin state.

Latch-Up Protection

GAL devices are designed with an on-chip charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the circuitry to latch. Additionally, outputs are designed with n-channel pullups instead of the traditional p-channel pullups to eliminate any possibility of SCR induced latching.

Manufacturer Testing

Because of E^{2}CMOS technology, GAL devices can be reprogrammed in milliseconds. This allows each device to be completely tested by the manufacturer using numerous logic array and architecture patterns prior to shipping. Every programmable cell and every logic path through every device is fully tested for programmability, functionality and performance to all AC and DC parameters. The customer can therefore expect 100% programming and functional yield and 100% compliance of all GAL products to datasheet specifications.

The testing procedure performed on all GAL devices by the manufacturer tests all aspects of device operation. Extensive testing of all programmable cells in the device include margin testing, internal verify, and program retention during high-temperature bake. All DC and AC parameters are tested at hot and cold temperatures using a variety of worst case logic and signal patterns. Functional tests include reprogramming each OLMC to all valid architectural configurations.

Register Preload

The register preload feature allows OLMC registers to be directly loaded with any desired data pattern. It also allows the present state of OLMC registers to be examined regardless of TRI-STATE control conditions. This simplifies testing of devices after programming. A device may be put into any desired register state at any point during the functional test sequence. The test sequence may then be resumed to verify proper next-state transitions. This allows complete verification of sequential logic circuits, including states that are normally impossible or difficult to reach. It may also shorten the overall test time significantly.

Register preload is not an operational mode and is not intended for board-level testing because elevated voltage levels must be applied to the device. The programming equipment normally provides the register preload capability as part of its functional test facility. Note that the testing of GAL devices after programming by the user may be considered unnecessary because all E^{2}CMOS GAL products are completely tested by the manufacturer, guaranteeing 100% post-programming functional yield.

The register preload algorithm is described for those users who wish to test programmed GAL devices using test equipment other than approved GAL programming equipment. As shown in the Register Preload Waveform in *Figure 5*, the preload sequence must not begin until the normal power-up reset operation has completed (after time t_{RESET}). The device is placed into preload mode by raising the "PRLD" input (pin* 11) to voltage V_{IES}, as specified in the Register Preload Specifications (Table III).

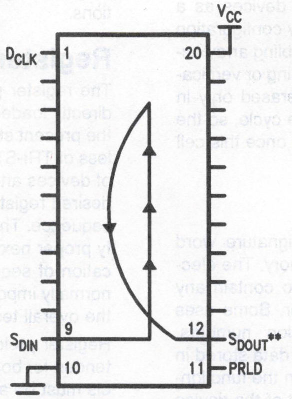
To preload the OLMC registers, a series of data bits are shifted into the device on the "S_{DIN}" input (pin* 9), one bit for each OLMC in which registered output has been selected. (Non-registered OLMCs are bypassed.) The shift sequence is clocked by the rising edge of the "D_{CLK}" input (pin* 1). The data stream is shifted in through the registered OLMC with the lowest corresponding pin number, and then "upward" through all remaining registered OLMCs in pin-number ascending order. Therefore, the first data bit in the series is ultimately loaded into the registered OLMC with the highest corresponding pin number, as shown in *Figure 4*.

As the data series is shifted into the S_{DIN} input, the contents of all registers (in registered OLMCs) are shifted "upward" and out onto the "S_{DOUT}" output (pin* 12). Complete present-state information can be examined in this manner. Test fixtures can be devised to test several GAL devices in which the S_{DOUT} pin of each chip is connected to the S_{DIN} pin of the next, and all preload and present-state data can be shifted around a single serial loop.

Note that when shifting register data into S_{DIN} or out of S_{DOUT}, V_{IL}/V_{OL} = register reset (0), and V_{IH}/V_{OH} = register set (1). These 0 and 1 register states are always inverted (active-low) on the normal output pins regardless of the selected output polarity (polarity affects logic function values before register inputs).

*Applies to both 20-pin DIP and 20-lead PCC Packages for GAL16V8.

Register Preload (Continued)


VCC
DCLK
1
20
9
SDIN
12
SDOUT**
10
11
PRLD

TL/L/9344-17

**The S_{DOUT} output buffer is an open drain output during preload. This pin should be terminated to V_{CC} with a 10 kΩ resistor.

FIGURE 4. Output Register Preload Pinout

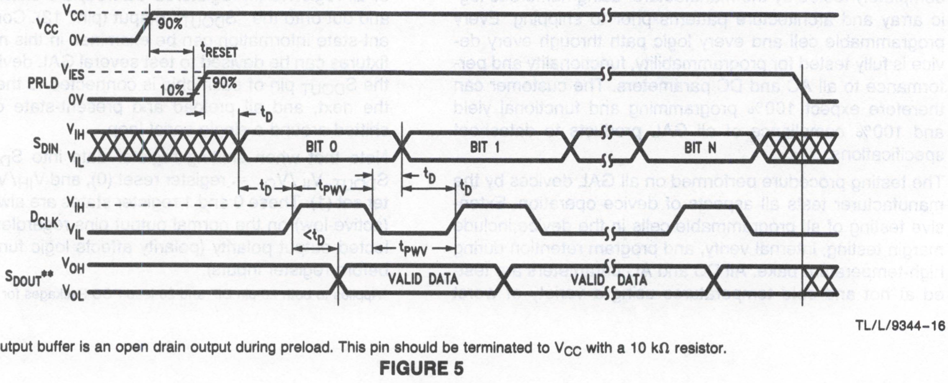
Register Preload Specifications

TABLE III

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	Input Voltage (High)		2.40		V_{CC}	V
V_{IL}	Input Voltage (Low)		0.00		0.50	V
V_{IES}	Registered Preload Input Voltage		14.5	15	15.5	V
V_{OH}	Output Voltage (High) (Note 1)				V_{CC}	V
V_{OL}	Output Voltage (Low) (Note 1)	$I_{OL} \leq 12$ mA	0.00		0.50	V
I_{IH}, I_{IL}	Input Current (Programming)			±1	±10	μA
I_{OH}	High Level Output Current (Note 1)	$V_{OH} \leq V_{CC}$			10	μA
t_{PWV}	Verify Pulse Width		1	5	10	μs
t_D	Pulse Sequence Delay		1	5	10	μs
t_{RESET}	Register Reset Time from Valid V_{CC}				45	μs

Note 1: The S_{DOUT} output buffer is an open drain output. This pin should be terminated to V_{CC} with a 10k resistor.

Register Preload Waveforms


VCC
VCC
0V
90%
tRESET
PRLD
VIES
0V
10%
90%
tD
SDIN
VIH
VIL
BIT 0
BIT 1
BIT N
tD
tD
tPWV
tD
DCLK
VIH
VIL
tD
tPWV
SDOUT**
VOH
VOL
VALID DATA
VALID DATA

TL/L/9344-16

**The S_{DOUT} output buffer is an open drain output during preload. This pin should be terminated to V_{CC} with a 10 kΩ resistor.

FIGURE 5

Programming Details

Understanding the information in this section is not essential when using approved programming equipment and software for developing GAL designs. This is a more thorough disclosure of the GAL architecture provided for direct JEDEC cell-map editing and diagnostic purposes. This section alone, however, does not contain sufficient information to implement the GAL programming algorithm. If detailed specifications of the GAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support department.

As mentioned in the Functional Description, the OLMC is responsible for selecting input and/or output paths, registered vs. combinatorial outputs, active-high or low polarity, and common vs. locally-controlled TRI-STATE control. Additionally, the OLMCs select between alternate logic array input paths to maintain JEDEC cell-map compatibility with either "small-PAL" or "medium-PAL" logic arrays.

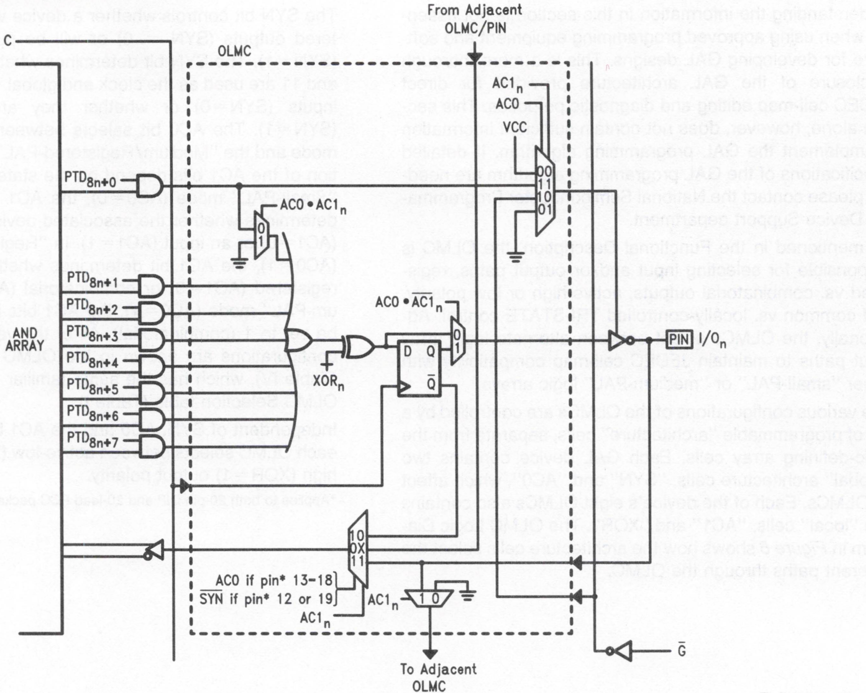
The various configurations of the OLMCs are controlled by a set of programmable "architecture" cells, separate from the logic-defining array cells. Each GAL device contains two "global" architecture cells, "SYN" and "AC0", which affect all OLMCs. Each of the device's eight OLMCs also contains two "local" cells, "AC1" and "XOR". The OLMC Logic Diagram in *Figure 6* shows how the architecture cells select the different paths through the OLMC.

The SYN bit controls whether a device will have any registered outputs (SYN = 0) or will be purely combinatorial (SYN = 1). The SYN bit determines whether device pins* 1 and 11 are used as the clock and global TRI-STATE control inputs (SYN = 0) or whether they are ordinary inputs (SYN = 1). The AC0 bit selects between the "Small-PAL" mode and the "Medium/Registered-PAL" modes. The function of the AC1 bits depend on the state of the AC0 bit. In "Small-PAL" mode (AC0 = 0), the AC1 bit in each OLMC determines whether the associated device pin is an output (AC1 = 0) or an input (AC1 = 1). In "Registered-PAL" mode (AC0 = 1), the AC1 bit determines whether each OLMC is registered (AC1 = 0) or combinatorial (AC1 = 1). In "Medium-PAL" mode (AC0 = 1), the AC1 bits in all OLMCs must be set to 1 (combinatorial). All of the valid architecture bit configurations are shown in the OLMC Architecture table (Table IV), which has the same familiar format used in the OLMC Selection table (Table I).

Independent of SYN, AC0 and the AC1 bits, the XOR bit in each OLMC selects between active-low (XOR = 0) or active-high (XOR = 1) output polarity.

*Applies to both 20-pin DIP and 20-lead PCC packages for GAL16V8.

OLMC Logic Diagram


From Adjacent OLMC/PIN
C
OLMC
AC1n
AC0
VCC
00
11
10
01
PTD8n+0
AC0 • AC1n
0
1
PTD8n+1
PTD8n+2
AND ARRAY
PTD8n+3
PTD8n+4
PTD8n+5
PTD8n+6
PTD8n+7
XORn
AC0 • AC1n
D Q
Q̄
0
1
PIN
I/On
10
0X
11
AC0 if pin* 13–18
SYN if pin* 12 or 19
AC1n
AC1n
1 0
G̅
To Adjacent OLMC
TL/L/9344–35

*Applies to both 20-pin DIP and 20-lead PCC packages for GAL16V8.

FIGURE 6

OLMC Architecture Programming

TABLE IV

	"Small-PAL" Mode			"Registered-PAL" Mode			"Medium-PAL" Mode	
	Function		JEDEC Input Line #s (Note 1)	Function		JEDEC Input Line #s (Note 1)	Function	JEDEC Input Lines #s (Note 1)
Pin 1	INPUT	INPUT	2,3	CLOCK	CLOCK		INPUT	2,3
*** Pin 19	OUTPUT*	INPUT	6,7	REGISTER	I/O	2,3	TRI-STATE**	
*** Pin 18	OUTPUT*	INPUT	10,11	REGISTER	I/O	6,7	I/O	6,7
*** Pin 17	OUTPUT*	INPUT	14,15	REGISTER	I/O	10,11	I/O	10,11
*** Pin 16	OUTPUT*	NC		REGISTER	I/O	14,15	I/O	14,15
*** Pin 15	OUTPUT*	NC		REGISTER	I/O	18,19	I/O	18,19
*** Pin 14	OUTPUT*	INPUT	18,19	REGISTER	I/O	22,23	I/O	22,23
*** Pin 13	OUTPUT*	INPUT	22,23	REGISTER	I/O	26,27	I/O	26,27
*** Pin 12	OUTPUT*	INPUT	26,27	REGISTER	I/O	30,13	TRI-STATE**	
Pin 11	INPUT	INPUT	30,31	$\overline{G}$	$\overline{G}$		INPUT	30,31
	AC1$_n$ = 0	AC1$_n$ = 1		AC1$_n$ = 0	AC1$_n$ = 1		AC1$_n$ = 1	
	SYN = 1, AC0 = 0			SYN = 0, AC0 = 1			SYN = 1, AC0 = 1	
	All outputs are combinatorial and always active.			At least one output is registered.			All I/O pins are combinatorial.	

Note: Pin numbers above apply to both 20-pin DIP and 20-lead PCC packages for GAL16V8.

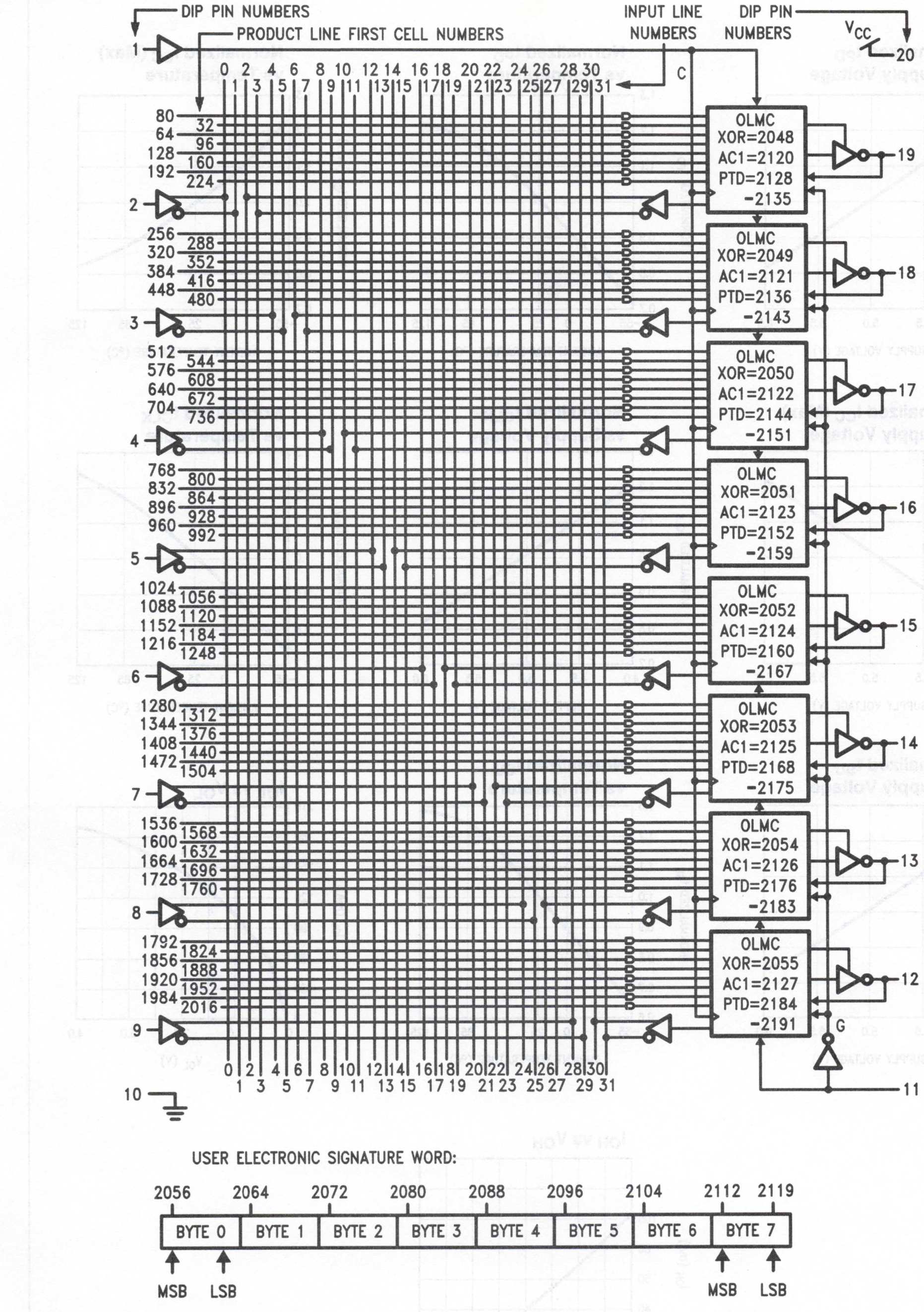
Note 1: All even and odd numbered JEDEC input line numbers correspond to true and complement array inputs, respectively.

*Active combinatorial output.

**TRI-STATE combinatorial output.

*** AC1$_n$ applies to these I/O pins only.

GAL16V8 Logic Diagram

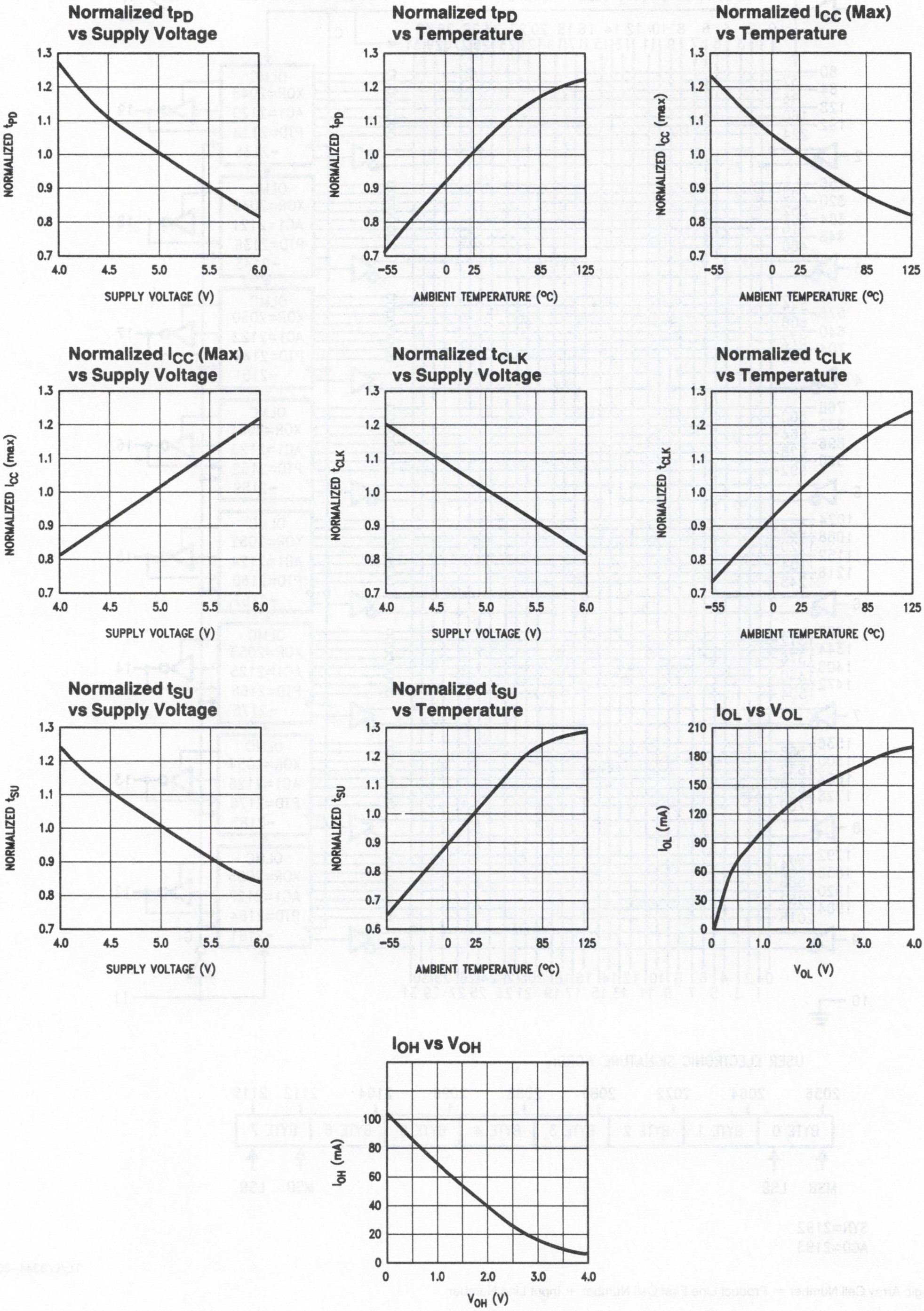

DIP PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
C
V_{CC}
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1
80 32 64 96 128 160 192 224
2
256 288 320 352 384 416 448 480
3
512 544 576 608 640 672 704 736
4
768 800 832 864 896 928 960 992
5
1024 1056 1088 1120 1152 1184 1216 1248
6
1280 1312 1344 1376 1408 1440 1472 1504
7
1536 1568 1600 1632 1664 1696 1728 1760
8
1792 1824 1856 1888 1920 1952 1984 2016
9
10
OLMC XOR=2048 AC1=2120 PTD=2128 -2135
19
OLMC XOR=2049 AC1=2121 PTD=2136 -2143
18
OLMC XOR=2050 AC1=2122 PTD=2144 -2151
17
OLMC XOR=2051 AC1=2123 PTD=2152 -2159
16
OLMC XOR=2052 AC1=2124 PTD=2160 -2167
15
OLMC XOR=2053 AC1=2125 PTD=2168 -2175
14
OLMC XOR=2054 AC1=2126 PTD=2176 -2183
13
OLMC XOR=2055 AC1=2127 PTD=2184 -2191
12
G
11
USER ELECTRONIC SIGNATURE WORD:
2056 2064 2072 2080 2088 2096 2104 2112 2119
BYTE 0 BYTE 1 BYTE 2 BYTE 3 BYTE 4 BYTE 5 BYTE 6 BYTE 7
MSB LSB MSB LSB
SYN=2192
ACO=2193

TL/L/9344-36

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

FIGURE 7

Typical Performance Characteristics



TL/L/9344–19

National Semiconductor

GAL20V8 Generic Array Logic

General Description

The NSC E²CMOS™ GAL® device combines a high performance CMOS process with electrically erasable floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels.

The 24-pin GAL20V8 features 8 programmable Output Logic Macrocells (OLMCs) allowing each TRI-STATE® output to be configured by the user. Additionally, the GAL20V8 is capable of emulating, in a functional/fuse map/parametric compatible device, the most popular 24-pin PAL® device architectures.

Programming is accomplished using readily available hardware and software tools. NSC guarantees a minimum 100 erase/write cycles.

Unique test circuitry and reprogrammable cells allow complete AC, DC, cell and functionality testing during manufacture. Therefore, NSC guarantees 100% field programmability and functionality of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.

Features

- Electrically erasable cell technology
  - — Reconfigurable logic
  - — Reprogrammable cells
  - — Guaranteed 100% yields
- High performance E²CMOS technology
  - — Low power: 45 mA/90 mA max active
  - — High speed: 20 ns–35 ns max access
- Eight output logic macrocells
  - — Maximum flexibility for complex logic designs
  - — Also emulates 24-pin PAL devices with full function/fuse map/parametric compatibility
- Preload and power-on reset of all registers
  - — 100% functional testability
- Fully supported by National PLAN™ development software
- High speed programming algorithm
- Security cell prevents copying logic

PAL Replacement by Device Type

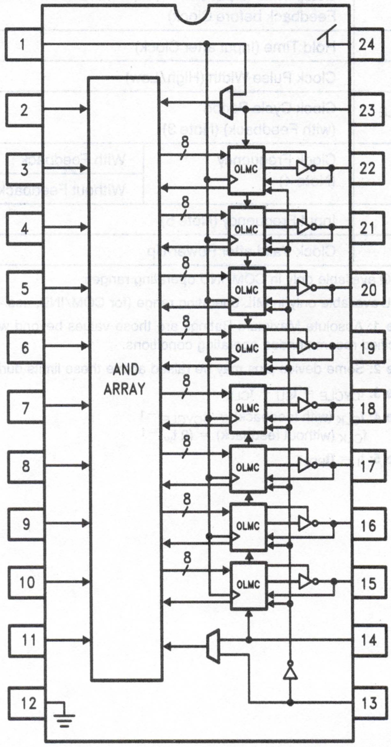
"Small-PAL" Mode				"Registered-PAL" Mode			"Medium-PAL" Mode
14L8	16L6	18L4	20L2	20R8	20R6	20R4	20L8
14H8	16H6	18H4	20H2	20RP8	20RP6	20RP4	20H8
14P8	16P6	18P4	20P2				20P8

PAL Replacement by Speed/Power

PAL Rev	Speed	Power*	GAL Rev
B-2	25 ns	105 mA	25L or 20L (90 mA) or 25Q (45 mA)
A	25 ns	210 mA	25L or 20L (90 mA)
B-2 MIL	30 ns	105 mA	30L or 25L (110 mA) or 30Q (55 mA)
A MIL	30 ns	210 mA	30L or 25L (110 mA)
B-4	35 ns	55 mA	30Q (45 mA)
A-2	35 ns	105 mA	30Q (45 mA)
STD	35 ns	210 mA	25L (90 mA)

*Shown for Medium PAL products.

Block Diagram—GAL20V8



TL/L/9343-21

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−2.5V to V_{CC} +1.0V
Off-State Output Voltage (Note 2)	−2.5V to V_{CC} +1.0V
Output Current	+100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	260°C
ESD Tolerance	500V

C_{ZAP} = 100 pF
R_{ZAP} = 150Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

SUPPLY VOLTAGE AND TEMPERATURE

Symbol	Parameter	Commercial			Industrial			Military			Units
		Min	Nom	Max	Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	4.5	5	5.5	4.5	5	5.5	V
T_A	Operating Free-Air Temperature	0	25	75	−40	25	85	−55	25		°C
T_C	Operating Case Temperature									125	°C

AC TIMING REQUIREMENTS

Symbol	Parameter		GAL20V8-20L COM/IND		GAL20V8-25Q GAL20V8-25L COM/IND MIL*		GAL20V8-30Q GAL20V8-30L COM/IND* MIL		Units
			Min	Max	Min	Max	Min	Max	
t_{SU}	Setup Time (Input or Feedback before Clock)		15		20		25		ns
t_H	Hold Time (Input after Clock)		0		0		0		ns
t_W	Clock Pulse Width (High/Low)		12		15		15		ns
t_{CYCLE}	Clock Cycle Period (with Feedback) (Note 3)		30		35		45		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		33.3		28.5		22.2	MHz
		Without Feedback		41.6		33.3		33.3	
f_I	Input Frequency (Note 5)			50.0		40.0		33.3	
t_{PR}	Clock Valid after Power-Up			100		100		100	ns

*-25Q available only in COM/IND operating ranges;
-30L available only in MIL operating range (for COM/IND, use -25L or -30Q).

Note 1: Absolute Maximum Ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: $t_{CYCLE} = t_{SU} + t_{CLK}$

Note 4: f_{CLK} (with feedback) = $(t_{CYCLE})^{-1}$
f_{CLK} (without feedback) = $(2\ t_w)^{-1}$

Note 5: $t_I = (t_{PD})^{-1}$

Electrical Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Conditions	Temperature Range	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage			2.0		$V_{CC}+1$	V
V_{IL}	Low Level Input Voltage			−1.0		0.8	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, I_{OH} = −3.2 mA	COM/IND	2.4			V
		V_{CC} = Min, I_{OH} = −2.0 mA	MIL	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, I_{OL} = 24 mA	COM/IND			0.5	V
		V_{CC} = Min, I_{OL} = 12 mA	MIL			0.5	V
I_{OZH}	High Level Off-State Output Current	V_{CC} = Max, V_O = V_{CC} (Max)				10	µA
I_{OZL}	Low Level Off-State Output Current	V_{CC} = Max, V_O = GND				−10	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = V_{CC} (Max)				10	µA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = V_{CC} (Max)				10	µA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = GND				−10	µA
I_{OS}*	Output Short Circuit Current	V_{CC} = 5.0V, V_O = GND		−30		−130	mA
I_{CC}	Supply Current Quarter Power (GAL20V8-25Q, -30Q)	f = 15 MHz, V_{CC} = Max	COM			45	mA
			MIL/IND			55	mA
	Supply Current Half Power (GAL20V8-20L, -25L, -30L)	f = 15 MHz, V_{CC} = Max	COM			90	mA
			MIL/IND			110	mA
C_I	Input Capacitance	V_{CC} = 5.0V, V_I = 2.0V				12	pF
$C_{I/O}$	I/O Capacitance	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V				15	pF

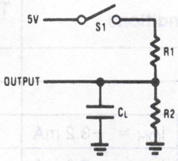
*One output at a time for a maximum duration of one second @ 25°C.

Switching Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Conditions	GAL20V8-20L		GAL20V8-25Q GAL20V8-25L		GAL20V8-30Q GAL20V8-30L		Units
			COM/IND		COM/IND MIL*		COM/IND* MIL		
			Min	Max	Min	Max	Min	Max	
t_{PD}	Input or Feedback to Combinatorial Output	S1 Closed, C_L = 50 pF		20		25		30	ns
t_{CLK}	Clock to Registered Output or Feedback	S1 Closed, C_L = 50 pF		15		15		20	ns
t_{PZXG}	$\overline{G}$ ↓ to Registered Output Enabled	Active High: S1 Open, C_L = 50 pF Active Low: S1 Closed C_L = 50 pF		18		20		25	ns
t_{PXZG}	$\overline{G}$ ↑ to Registered Output Disabled	From V_{OH}: S1 Open, C_L = 5 pF From V_{OL}: S1 Closed C_L = 5 pF		18		20		25	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	Active High: S1 Open, C_L = 50 pF Active Low: S1 Closed C_L = 50 pF		20		25		30	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	From V_{OH}: S1 Open, C_L = 5 pF From V_{OL}, S1 Closed C_L = 5 pF		20		25		30	ns
t_{RESET}	Power-Up to Registered Output High	S1 Closed C_L = 50 pF		45		45		45	µs

* -25Q available only in COM/IND operating ranges;
-30L available only in MIL operating range (for COM/IND, use -25L or -30Q).

AC Test Load


5V
S1
R1
OUTPUT
CL
R2

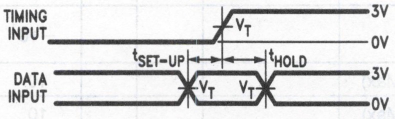
MIL
R1 = 390
R2 = 750

COM'L/IND
R1 = 200
R2 = 390

TL/L/9343-34

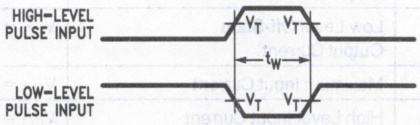
Test Waveforms

Setup and Hold


TIMING INPUT
VT
3V
0V
tSET-UP
tHOLD
DATA INPUT
VT
VT
3V
0V

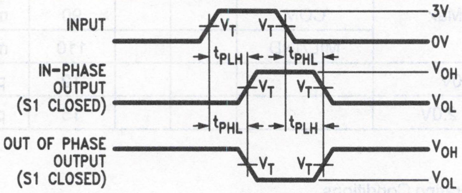
TL/L/9343-36

Pulse Width


HIGH-LEVEL PULSE INPUT
VT
VT
tW
LOW-LEVEL PULSE INPUT
VT
VT

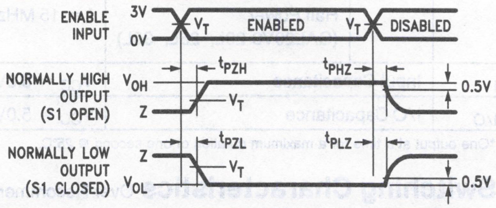
TL/L/9343-37

Propagation Delay


INPUT
VT
VT
3V
0V
tPLH
tPHL
IN-PHASE OUTPUT (S1 CLOSED)
VT
VT
VOH
VOL
tPHL
tPLH
OUT OF PHASE OUTPUT (S1 CLOSED)
VT
VT
VOH
VOL

TL/L/9343-38

Enable and Disable


ENABLE INPUT
3V
0V
VT
ENABLED
VT
DISABLED
tPZH
tPHZ
NORMALLY HIGH OUTPUT (S1 OPEN)
VOH
Z
VT
0.5V
tPZL
tPLZ
NORMALLY LOW OUTPUT (S1 CLOSED)
Z
VOL
VT
0.5V

TL/L/9343-39

Notes:

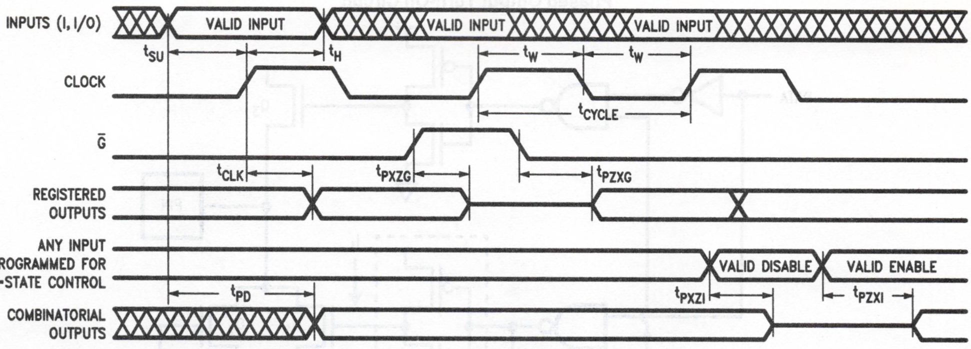
C_L includes probe and jig capacitance.

$V_T = 1.5V$.

Test inputs have rise and fall times of 5 ns between 0.3V and 2.7V.

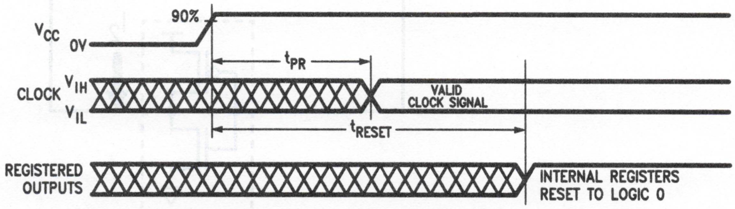
In the examples above, the phase relationships between inputs and outputs have been chosen arbitrarily.

Switching Waveforms


INPUTS (I, I/O)
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
t_{CYCLE}
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/9343-7

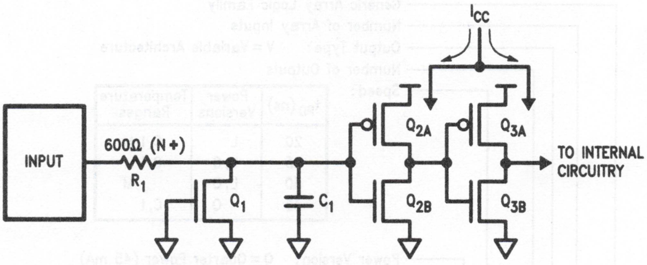
Power-Up Reset Waveforms


90%
V_{CC}
0V
t_{PR}
CLOCK
V_{IH}
V_{IL}
VALID CLOCK SIGNAL
t_{RESET}
REGISTERED OUTPUTS
INTERNAL REGISTERS RESET TO LOGIC 0

TL/L/9343-42

Input/Output Schematics

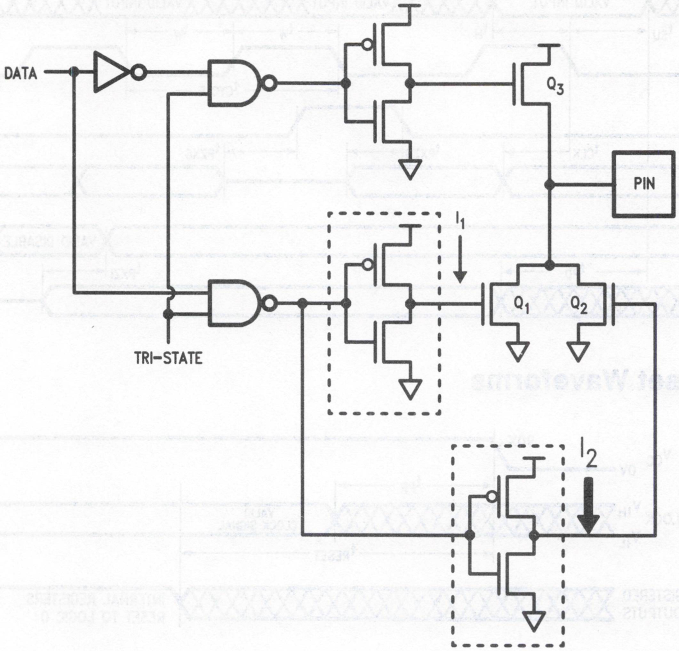
Input Translator/Buffer


I_{CC}
INPUT
600Ω (N+)
R_1
Q_1
C_1
Q_{2A}
Q_{2B}
Q_{3A}
Q_{3B}
TO INTERNAL CIRCUITRY

TL/L/9343-40

Input/Output Schematics (Continued)

Phased Output Turn-On Circuit



TL/L/9343-41

Ordering Information

GAL 20 V 8 − 20 L N C

- Generic Array Logic Family
- Number of Array Inputs
- Output Type: V = Variable Architecture
- Number of Outputs
- Speed:

t_{PD} (ns)	Power Versions	Temperature Ranges
20	L	C, I, M
25	L, Q	C, I
30	L, Q	M
35	Q	C, I

- Power Version: Q = Quarter Power (45 mA)
  L = Half Power (90 mA)
- Package Type: N = 24-Pin Plastic DIP
  J = 24-Pin Ceramic DIP
  V = 28-Lead Plastic Chip Carrier
- Temperature Range: C = Commercial (0°C to +75°C)
  I = Industrial (−40°C to +85°C)
  M = Military (−55°C to +125°C)

TL/L/9343-43

Functional Description

The GAL logic array consists of a programmable AND array with fixed OR-gate connections, similar to the bipolar PAL architecture. The logic array is organized as 20 complementary input lines crossing 64 "product term" lines with a programmable E^{2}PROM cell at each intersection (2560 cells). Each programmable cell may establish a connection between an input line (true or complement phase of an array input signal) and a product term. A product term is satisfied (logically true) while all of the input lines "connected" to it are in the high logic state.

The 64 product terms are organized into eight output groups with eight terms each. Seven or eight of the product terms in each output group feed into an OR-gate to produce each output logic function; one of the product terms may instead be used to control the associated TRI-STATE device output. The fundamental transfer function of each GAL output is the familiar Boolean sum-of-products. Design development software is available which accepts Boolean equations and converts them automatically into GAL programming patterns.

As shown in the GAL20V8 Block Diagram *(Figure 1)*, a total of eight output logic functions are available. Each of the AND/OR logic functions feeds into an "output logic macrocell" (OLMC). The eight OLMCs control the flow of input and output signals between the logic array and the device's I/O pins.

Under control of an OLMC, each output may be designated either registered or combinatorial (non-registered). In the registered output configuration, the logic function output passes through a D-type flip-flop triggered by the rising edge of the clock input. Additionally, the logic function's output polarity may be designated active-low or active-high (adjusted before the register, if present). OLMC options such as these are selected using a set of programmable architecture control cells. These architecture cells are normally configured automatically by the development software or programming hardware.

All of the possible I/O configurations of the GAL20V8 are classified into three basic modes: "Small-PAL" mode, "Registered-PAL" mode and "Medium-PAL" mode. These modes correspond to the architectures of the PAL families which the GAL20V8 can emulate. The modes determine the mixture of OLMC configurations which can be selected for the device. The OLMC Selection table (Table I) lists which functions can be selected on device pins* 1, 13 and 15 through 22 for each of the three modes. The logic diagrams in *Figure 3* illustrate these OLMC functions.

"OUTPUT" represents the always-active combinatorial output configuration available in the "Small-PAL" mode. "REGISTER" is the registered output with register feedback available in the "Registered-PAL" mode. "I/O" is the combinatorial bidirectional I/O available in "Registered-PAL" and "Medium-PAL" modes. "TRI-STATE" is the TRI-STATE combinatorial output function appearing on pins* 15 and 22 in the "Medium-PAL" mode. "INPUT" in Table I denotes an OLMC used as a dedicated input only.

In the "Small-PAL" and "Medium-PAL" modes (Table I), pins* 1 and 13 are always dedicated inputs. In the "Registered-PAL" mode, however, pin* 1 becomes the clock input controlling all OLMC registers, and pin* 13 becomes the output enable ($\overline{G}$) input controlling the TRI-STATE outputs of all registered OLMCs. Within the "Small-PAL" and "Registered-PAL" modes in Table I, the functions of pins* 15 through 22 can be selected individually from either of the two functions listed. For example, in "Registered-PAL" mode, pins* 15 through 22 can each be designated as either a registered output or a combinatorial I/O. The "Medium-PAL" mode represents a single fixed configuration used to emulate combinatorial medium PAL devices (20L8, 20H8, 20P8).

Table II lists the bipolar PAL products which the GAL20V8 can emulate, and the specific input/output configurations used. This is just a subset, however, of all the configurations provided in Table I.

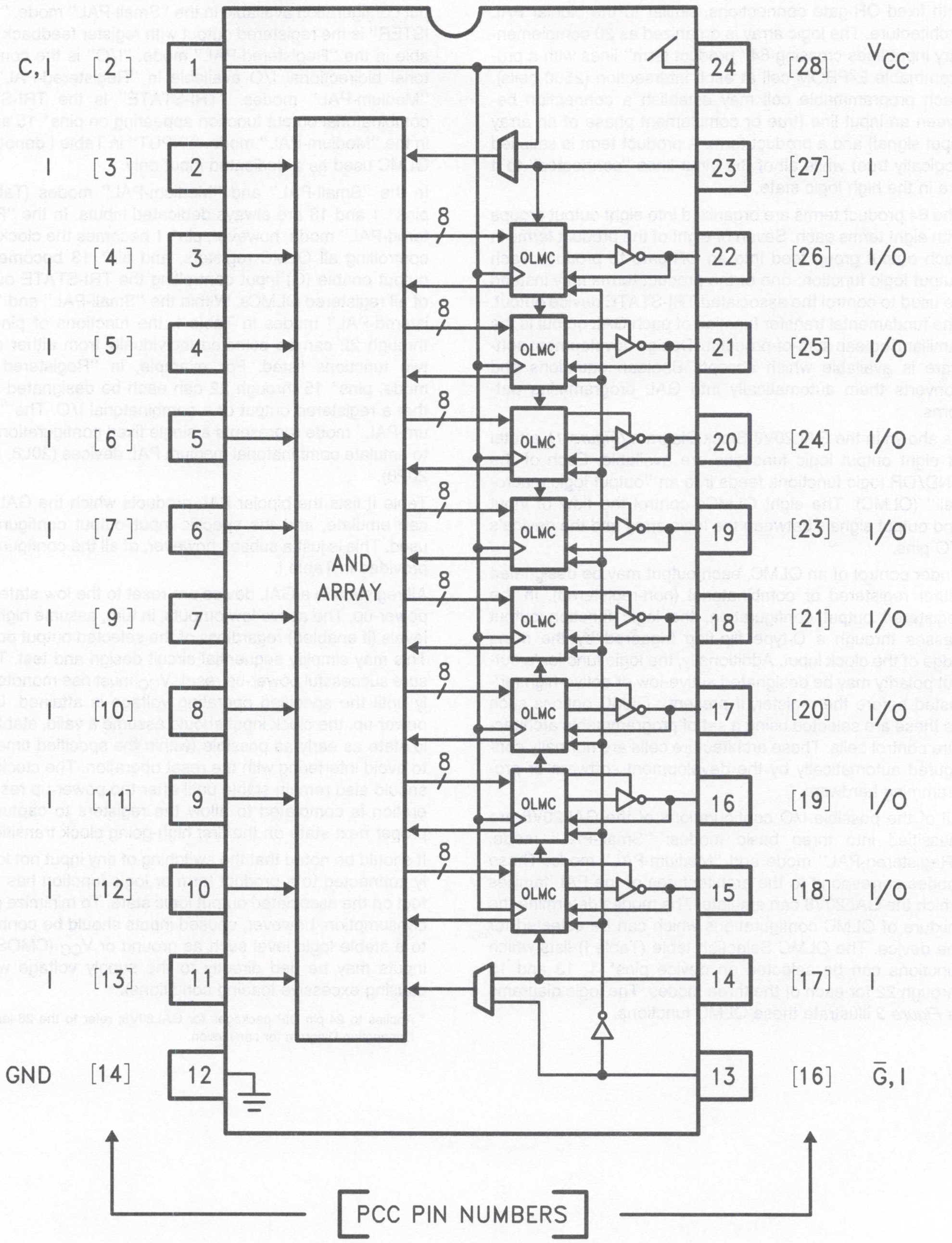
All registers in a GAL device are reset to the low state upon power-up. The active-low outputs, in turn, assume high logic levels (if enabled) regardless of the selected output polarity. This may simplify sequential circuit design and test. To ensure successful power-up reset, V_{CC} must rise monotonically until the specified operating voltage is attained. During power-up, the clock input should assume a valid, stable logic state as early as possible (within the specified time, t_{PR}) to avoid interfering with the reset operation. The clock input should also remain stable until after the power-up reset operation is completed to allow the registers to capture the proper next state on the first high-going clock transition.

It should be noted that the switching of any input not logically connected to a product term or logic function has no effect on the associated output logic state. To minimize power consumption, however, unused inputs should be connected to a stable logic level such as ground or V_{CC} (CMOS GAL inputs may be tied directly to the supply voltage without causing excessive loading conditions).

---

* Applies to 24-pin DIP packages for GAL20V8; refer to the 28-lead PCC Connection Diagram for conversion.

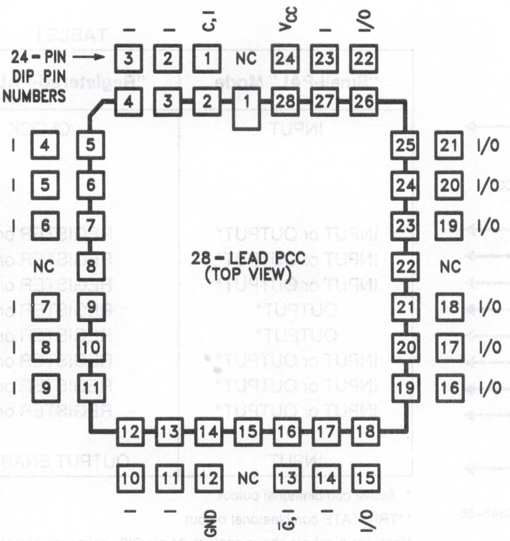
GAL20V8 Block Diagram—DIP Connections


C,I
[2]
1
I
[3]
2
I
[4]
3
I
[5]
4
I
[6]
5
I
[7]
6
I
[9]
7
I
[10]
8
I
[11]
9
I
[12]
10
I
[13]
11
GND
[14]
12
AND
ARRAY
24
[28]
V_CC
23
[27]
I
8
OLMC
22
[26]
I/O
8
OLMC
21
[25]
I/O
8
OLMC
20
[24]
I/O
8
OLMC
19
[23]
I/O
8
OLMC
18
[21]
I/O
8
OLMC
17
[20]
I/O
8
OLMC
16
[19]
I/O
8
OLMC
15
[18]
I/O
14
[17]
I
13
[16]
G̅,I
PCC PIN NUMBERS

TL/L/9343–44

FIGURE 1

28-Lead PCC Connection Diagram



TL/L/9343-45

FIGURE 2

Clock/Input Frequency Specifications

The clock frequency (f_{CLK}) parameter listed in the Recommended Operating Conditions table specifies the maximum speed at which the GAL registers are guaranteed to operate. Clock frequency is defined differently for the two cases in which register feedback is used versus when it is not. In a data-path type application, when the logic functions fed into the registers are not dependent on register feedback from the previous cycle (i.e. based only on external inputs), the minimum required cycle period (f_{CLK}^{-1} without feedback) is defined as the greater of the minimum clock period (t_w high + t_w low) and the minimum "data window" period ($t_{SU} + t_H$). This assumes optimal alignment between data inputs and the clock input. In sequential logic applications such as state machines, the minimum required cycle period ($t_{CYCLE} = f_{CLK}^{-1}$ with feedback) is defined as $t_{CLK} + t_{SU}$. This provides sufficient time for outputs from the registers to feed back through the logic array and set up on the inputs to the registers before the end of each cycle.

The input frequency (f_I) parameter specifies the maximum rate at which each GAL input can be toggled and still produce valid logic transitions on each combinatorial output. The f_I specification is derived as the inverse of the combinatorial propagation delay (t_{PD}).

Design Development Support

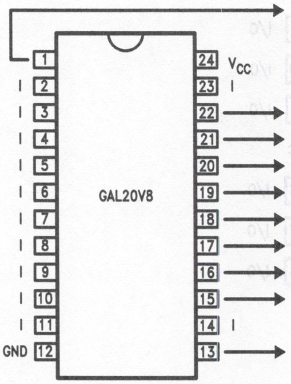
A variety of software tools and programming equipment is available to support the development of designs using GAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate a JEDEC-compatible "cell-map" (analogous to a PAL "fuse-map"). The industry-standard JEDEC format ensures that the resulting cell-map file can be down-loaded into a variety of programming equipment. Many software packages and programming units support a large variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

National strongly recommends using only approved programming hardware and software for developing GAL designs. Programming using unapproved equipment generally voids all guarantees. Approved programmers incorporate specialized programming algorithms that program the array and automatically configure the architecture cells. To ensure data retention and reliability, the programming algorithm also tracks the number of programming cycles to which each GAL device has been subjected since shipment, and stores this information automatically in the device.

The special GAL programming algorithm can also program a GAL device using a standard fuse-map developed for any of the emulated PAL products. PAL fuse-maps can be created by any JEDEC-compatible PAL development software or by loading the fuse pattern from an existing programmed PAL device into the programming unit (provided the PAL device has not been secured). However, to utilize the full flexibility of the GAL architecture, true GAL development software (such as PLAN software) is recommended.

Detailed logic diagrams showing all JEDEC cell-map addresses in the GAL logic array and OLMC are provided for direct map editing and diagnostic purposes (see "Programming Details"). For a list of current software and programming support tools available for these devices, please contact your local National sales representative or distributor. If detailed specifications of the GAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support department.

OLMC Selection Table



TL/L/9343-55

TABLE I

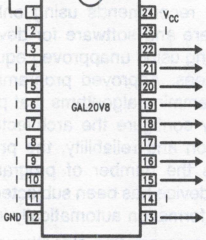
"Small-PAL" Mode	"Registered-PAL" Mode	"Medium-PAL" Mode
INPUT	CLOCK	INPUT
INPUT or OUTPUT*	REGISTER or I/O	TRI-STATE**
INPUT or OUTPUT*	REGISTER or I/O	I/O
INPUT or OUTPUT*	REGISTER or I/O	I/O
OUTPUT*	REGISTER or I/O	I/O
OUTPUT*	REGISTER or I/O	I/O
INPUT or OUTPUT*	REGISTER or I/O	I/O
INPUT or OUTPUT*	REGISTER or I/O	I/O
INPUT or OUTPUT*	REGISTER or I/O	TRI-STATE**
INPUT	OUTPUT ENABLE ($\overline{G}$)	INPUT

* Active combinatorial output

**TRI-STATE combinatorial output

Note: Pin numbers above apply to 24-pin DIP packages; refer to the 28-lead PCC Connection Diagram for conversion.

PAL Replacement Configurations



TL/L/9343-57

TABLE II

	"Small-PAL" Mode				"Registered-PAL" Mode			"Medium-PAL" Mode
	INPUT	INPUT	INPUT	INPUT	CLOCK	CLOCK	CLOCK	INPUT
	OUTPUT*	INPUT	INPUT	INPUT	REGISTER	I/O	I/O	TRI-STATE**
	OUTPUT*	OUTPUT*	INPUT	INPUT	REGISTER	REGISTER	I/O	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	INPUT	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	OUTPUT*	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	OUTPUT*	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	INPUT	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	INPUT	INPUT	REGISTER	REGISTER	I/O	I/O
	OUTPUT*	INPUT	INPUT	INPUT	REGISTER	I/O	I/O	TRI-STATE**
	INPUT	INPUT	INPUT	INPUT	$\overline{G}$	$\overline{G}$	$\overline{G}$	INPUT
Emulated PAL Products	14L8 14H8 14P8	16L6 16H6 16P6	18L4 18H4 18P4	20L2 20H2 20P2	20R8 20RP8	20R6 20RP6	20R4 20RP4	20L8 20H8 20P8

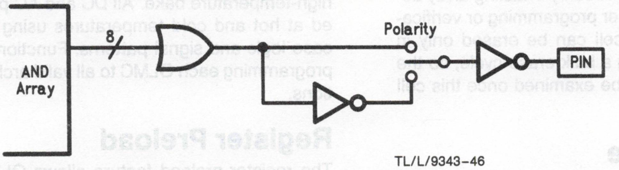
* Active combinatorial output.

**TRI-STATE combinatorial output.

Note: Pin numbers above apply to 24-pin DIP packages; refer to the 28-pin PCC Connection Diagram for conversion.

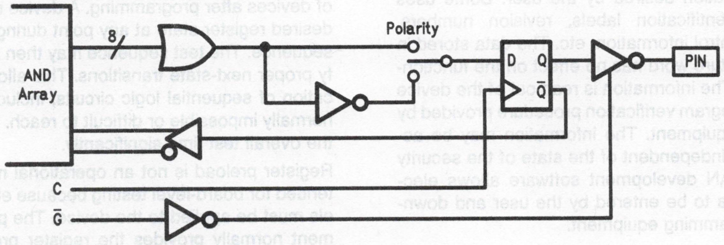
OLMC Configurations

OUTPUT (Active Combinatorial Output)


8
AND
Array
Polarity
PIN

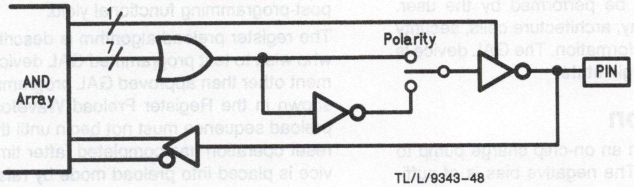
TL/L/9343-46

REGISTER (Registered Output)


8
AND
Array
Polarity
D
Q
Q̄
PIN
C
Ḡ

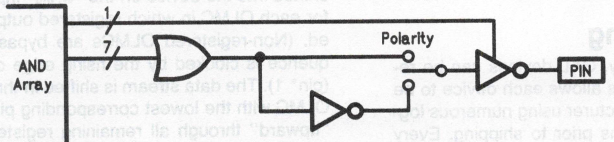
TL/L/9343-47

I/O (Combinatorial Input/Output)


1
7
AND
Array
Polarity
PIN

TL/L/9343-48

TRI-STATE (TRI-STATE Combinatorial Output)


1
7
AND
Array
Polarity
PIN

TL/L/9343-56

FIGURE 3

Security Cell

A security cell is provided on all GAL20V8 devices as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, the circuitry enabling array access is disabled, preventing further programming or verification of the array. The security cell can be erased only in conjunction with the array during a bulk erase cycle, so the original configuration can never be examined once this cell is programmed.

Electronic Signature

Each GAL device contains an electronic signature word consisting of 64 bits of reprogrammable memory. The electronic signature word can be programmed to contain any identification information desired by the user. Some uses include pattern identification labels, revision numbers, dates, inventory control information, etc. The data stored in the electronic signature word has no effect on the functionality of the device. The information is read out of the device using the normal program verification procedure provided by the programming equipment. The information may be accessed at any time independent of the state of the security cell. National's PLAN development software allows electronic signature data to be entered by the user and downloaded to the programming equipment.

Bulk Erase

The programming equipment automatically performs a bulk erase operation prior to each programming operation. No special erase operation need be performed by the user. Bulk erase clears the logic array, architecture cells, security cell, and electronic signature information. The GAL device is thereby reverted back to its virgin state.

Latch-Up Protection

GAL devices are designed with an on-chip charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the circuitry to latch. Additionally, outputs are designed with n-channel pullups instead of the traditional p-channel pullups to eliminate any possibility of SCR induced latching.

Manufacturer Testing

Because of E^{2}CMOS technology, GAL devices can be reprogrammed in milliseconds. This allows each device to be completely tested by the manufacturer using numerous logic array and architecture patterns prior to shipping. Every programmable cell and every logic path through every device is fully tested for programmability, functionality and performance to all AC and DC parameters. The customer can therefore expect 100% programming and functional yield and 100% compliance of all GAL products to datasheet specifications.

The testing procedure performed on all GAL devices by the manufacturer tests all aspects of device operation. Extensive testing of all programmable cells in the device include margin testing, internal verify, and program retention during high-temperature bake. All DC and AC parameters are tested at hot and cold temperatures using a variety of worst-case logic and signal patterns. Functional tests include reprogramming each OLMC to all valid architectural configurations.

Register Preload

The register preload feature allows OLMC registers to be directly loaded with any desired data pattern. It also allows the present state of OLMC registers to be examined regardless of TRI-STATE control conditions. This simplifies testing of devices after programming. A device may be put into any desired register state at any point during the functional test sequence. The test sequence may then be resumed to verify proper next-state transitions. This allows complete verification of sequential logic circuits, including states that are normally impossible or difficult to reach. It may also shorten the overall test time significantly.

Register preload is not an operational mode and is not intended for board-level testing because elevated voltage levels must be applied to the device. The programming equipment normally provides the register preload capability as part of its functional test facility. Note that the testing of GAL devices after programming by the user may be considered unnecessary because all E^{2}CMOS GAL products are completely tested by the manufacturer, guaranteeing 100% post-programming functional yield.

The register preload algorithm is described for those users who wish to test programmed GAL devices using test equipment other than approved GAL programming equipment. As shown in the Register Preload Waveform in *Figure 5,* the preload sequence must not begin until the normal power-up reset operation has completed (after time t_{RESET}). The device is placed into preload mode by raising the "PRLD" input (pin* 13) to voltage V_{IES}, as specified in the Register Preload Specifications (Table III).

To preload the OLMC registers, a series of data bits are shifted into the device on the "S_{DIN}" input (pin* 11), one bit for each OLMC in which registered output has been selected. (Non-registered OLMCs are bypassed.) The shift sequence is clocked by the rising edge of the "D_{CLK}" input (pin* 1). The data stream is shifted in through the registered OLMC with the lowest corresponding pin number, and then "upward" through all remaining registered OLMCs in pin-number ascending order. Therefore, the first data bit in the series is ultimately loaded into the registered OLMC with the highest corresponding pin number, as shown in *Figure 4.*

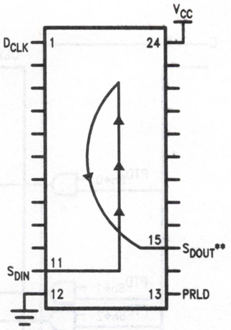
*Applies to 24-pin DIP packages for GAL20V8; refer to the 28-lead PCC Connection Diagram for conversion.

Register Preload (Continued)

As the data series is shifted into the S_{DIN} input, the contents of all registers (in registered OLMCs) are shifted "upward" and out onto the "S_{DOUT}" output (pin* 15). Complete present-state information can be examined in this manner. Test fixtures can be devised to test several GAL devices in which the S_{DOUT} pin of each chip is connected to the S_{DIN} pin of the next, and all preload and present-state data can be shifted around a single serial loop.

Note that when shifting register data into S_{DIN} or out of S_{DOUT}, V_{IL}/V_{OL} = register reset (0), and V_{IH}/V_{OH} = register set (1). These 0 and 1 register states are always inverted (active-low) on the normal output pins regardless of the selected output polarity (polarity affects logic function values before register inputs).

* Applies to 24-pin DIP packages for GAL20V8; refer to the 28-lead PCC Connection Diagram for conversion.



TL/L/9343-17

** The S_{DOUT} output buffer is an open drain output during preload. This pin should be terminated to V_{CC} with a 10 kΩ resistor.

FIGURE 4. Output Register Preload Pinout

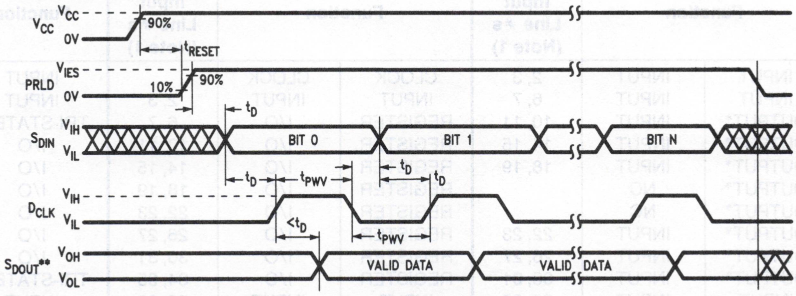
Register Preload Specifications

TABLE III

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	Input Voltage (High)		2.40		V_{CC}	V
V_{IL}	Input Voltage (Low)		0.00		0.50	V
V_{IES}	Register Preload Input Voltage		14.5	15	15.5	V
V_{OH}	Output Voltage (High) (Note 1)				V_{CC}	V
V_{OL}	Output Voltage (Low) (Note 1)	$I_{OL} \leq 12$ mA	0.00		0.50	V
I_{IH}, I_{IL}	Input Current (Programming)			±1	±10	μA
I_{OH}	High Level Output Current (Note 1)	$V_{OH} \leq V_{CC}$			10	μA
t_{PWV}	Verify Pulse Width		1	5	10	μs
t_D	Pulse Sequence Delay		1	5	10	μs
t_{RESET}	Register Reset Time from Valid V_{CC}				45	μs

Note 1: The S_{DOUT} output buffer is an open drain output. This pin should be terminated to V_{CC} with a 10k resistor.

Register Preload Waveforms

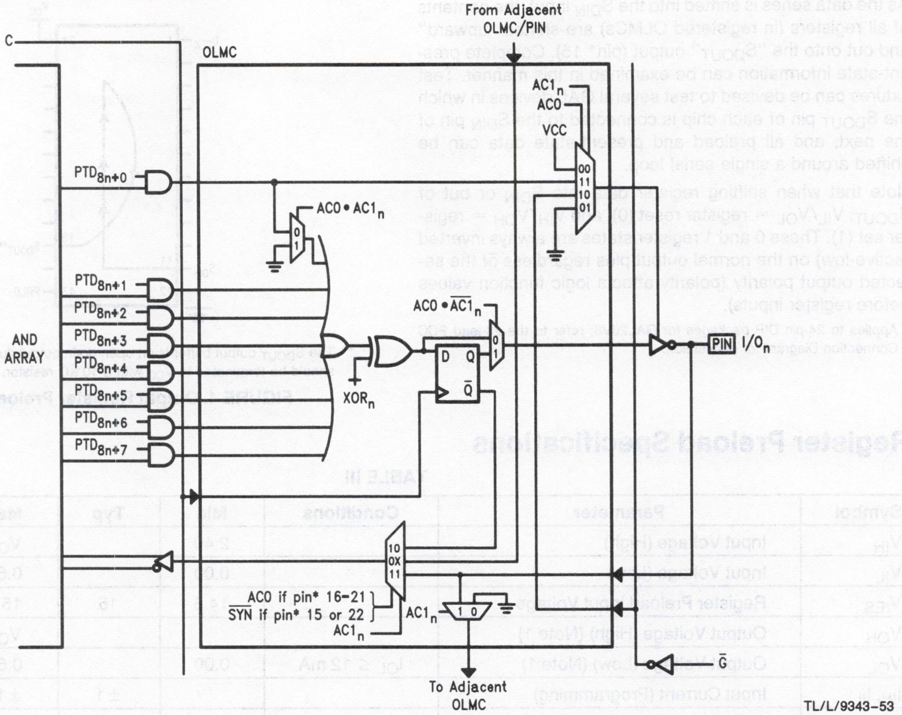


TL/L/9343-16

FIGURE 5

**The S_{DOUT} output buffer is an open drain output during preload. This pin should be terminated to V_{CC} with a 10 kΩ resistor.

OLMC Logic Diagram


From Adjacent OLMC/PIN
C
OLMC
AC1$_n$
AC0
VCC
00
11
10
01
PTD$_{8n+0}$
AC0 • AC1$_n$
0
1
PTD$_{8n+1}$
PTD$_{8n+2}$
PTD$_{8n+3}$
AND ARRAY
PTD$_{8n+4}$
PTD$_{8n+5}$
PTD$_{8n+6}$
PTD$_{8n+7}$
XOR$_n$
AC0 • $\overline{AC1}_n$
0
1
D
Q
$\overline{Q}$
PIN
I/O$_n$
10
0X
11
AC0 if pin* 16–21
SYN if pin* 15 or 22
AC1$_n$
AC1$_n$
1
0
$\overline{G}$
To Adjacent OLMC
TL/L/9343-53

*Applies to 24-pin DIP packages for GAL20V8; refer to the 28-lead PCC Connection Diagram for conversion.

FIGURE 6

OLMC Architecture Programming

TABLE IV

	"Small-PAL" Mode			"Registered-PAL" Mode			"Medium-PAL" Mode	
	Function		JEDEC Input Line #s (Note 1)	Function		JEDEC Input Line #s (Note 1)	Function	JEDEC Input Line #s (Note 1)
Pin 1	INPUT	INPUT	2, 3	CLOCK	CLOCK		INPUT	2, 3
Pin 23	INPUT	INPUT	6, 7	INPUT	INPUT	2, 3	INPUT	6, 7
***Pin 22	OUTPUT*	INPUT	10, 11	REGISTER	I/O	6, 7	TRI-STATE**	
***Pin 21	OUTPUT*	INPUT	14, 15	REGISTER	I/O	10, 11	I/O	10, 11
***Pin 20	OUTPUT*	INPUT	18, 19	REGISTER	I/O	14, 15	I/O	14, 15
***Pin 19	OUTPUT*	NC		REGISTER	I/O	18, 19	I/O	18, 19
***Pin 18	OUTPUT*	NC		REGISTER	I/O	22, 23	I/O	22, 23
***Pin 17	OUTPUT*	INPUT	22, 23	REGISTER	I/O	26, 27	I/O	26, 27
***Pin 16	OUTPUT*	INPUT	26, 27	REGISTER	I/O	30, 31	I/O	30, 31
***Pin 15	OUTPUT*	INPUT	30, 31	REGISTER	I/O	34, 35	TRI-STATE**	
Pin 14	INPUT	INPUT	34, 35	INPUT	INPUT	38, 39	INPUT	34, 35
Pin 13	INPUT	INPUT	38, 39	$\overline{G}$	$\overline{G}$		INPUT	38, 39
	AC1$_n$ = 0	AC1$_n$ = 1		AC1$_n$ = 0	AC1$_n$ = 1		AC1$_n$ = 1	
	SYN = 1, AC0 = 0			SYN = 0, AC0 = 1			SYN = 1, AC0 = 1	
	All outputs are combinatorial and always active.			At least one output is registered.			All I/O pins are combinatorial.	

Note: Pin numbers above apply to 24-pin DIP packages; refer to the 28-lead PCC Connection Diagram for conversion.

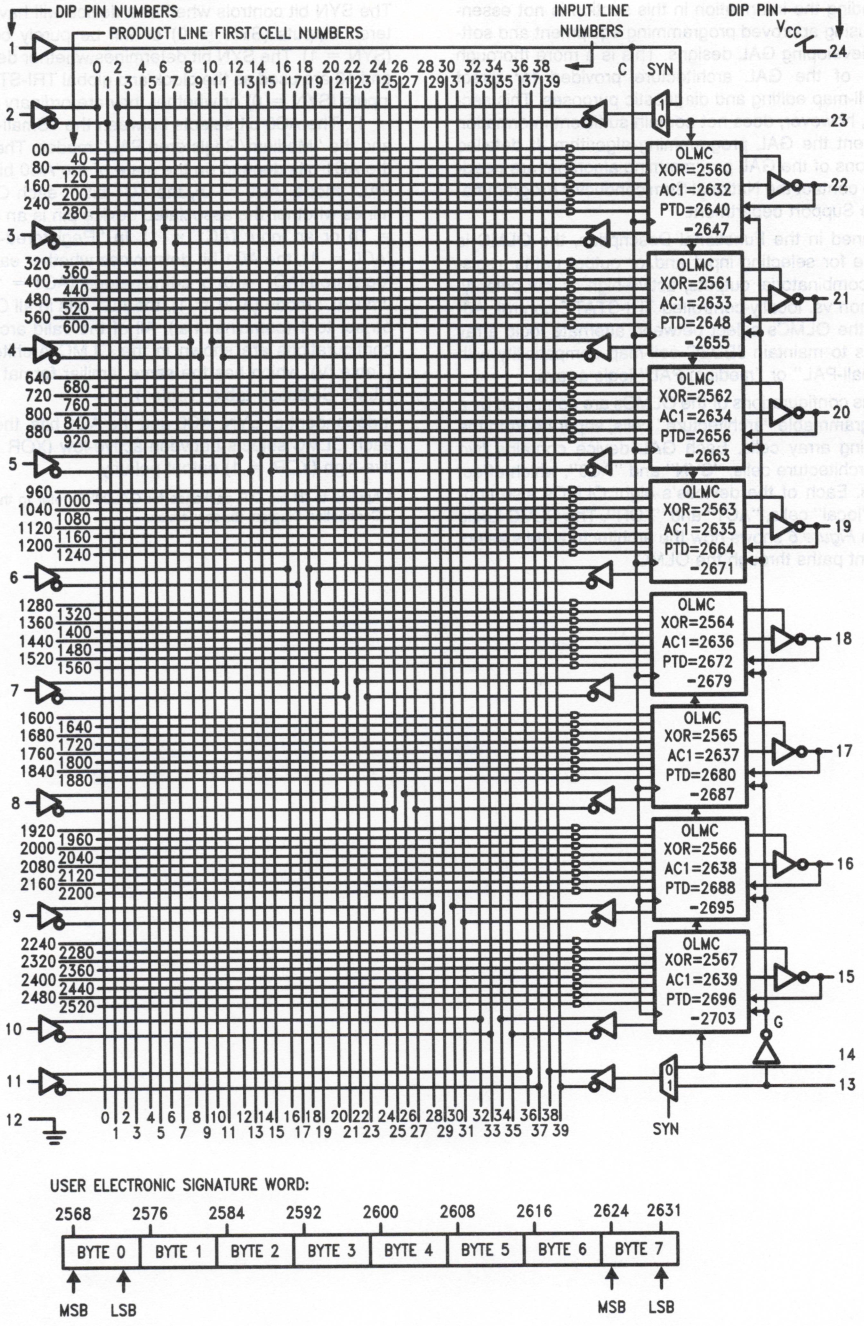
Note 1: All even and odd numbered JEDEC input line numbers correspond to true and complement array inputs, respectively.

*Active combinatorial output.

**TRI-STATE combinatorial output.

***AC1$_n$ applies to these I/O pins only.

GAL20V8 Logic Diagram


DIP PIN NUMBERS
PRODUCT LINE FIRST CELL NUMBERS
INPUT LINE NUMBERS
DIP PIN
VCC
24
C
SYN
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
1
2
3
4
5
6
7
8
9
10
11
12
23
22
21
20
19
18
17
16
15
14
13
G
00 40 80 120 160 200 240 280
320 360 400 440 480 520 560 600
640 680 720 760 800 840 880 920
960 1000 1040 1080 1120 1160 1200 1240
1280 1320 1360 1400 1440 1480 1520 1560
1600 1640 1680 1720 1760 1800 1840 1880
1920 1960 2000 2040 2080 2120 2160 2200
2240 2280 2320 2360 2400 2440 2480 2520
OLMC XOR=2560 AC1=2632 PTD=2640 -2647
OLMC XOR=2561 AC1=2633 PTD=2648 -2655
OLMC XOR=2562 AC1=2634 PTD=2656 -2663
OLMC XOR=2563 AC1=2635 PTD=2664 -2671
OLMC XOR=2564 AC1=2636 PTD=2672 -2679
OLMC XOR=2565 AC1=2637 PTD=2680 -2687
OLMC XOR=2566 AC1=2638 PTD=2688 -2695
OLMC XOR=2567 AC1=2639 PTD=2696 -2703
USER ELECTRONIC SIGNATURE WORD:
2568 2576 2584 2592 2600 2608 2616 2624 2631
BYTE 0 BYTE 1 BYTE 2 BYTE 3 BYTE 4 BYTE 5 BYTE 6 BYTE 7
MSB LSB
MSB LSB

SYN=2704
AC0=2705

TL/L/9343-54

JEDEC Logic Array Cell Number = Product Line First Cell Number + Input Line Number

FIGURE 7

Programming Details

Understanding the information in this section is not essential when using approved programming equipment and software for developing GAL designs. This is a more thorough disclosure of the GAL architecture provided for direct JEDEC cell-map editing and diagnostic purposes. This section alone, however, does not contain sufficient information to implement the GAL programming algorithm. If detailed specifications of the GAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support department.

As mentioned in the Functional Description, the OLMC is responsible for selecting input and/or output paths, registered vs. combinatorial outputs, active-high or low polarity, and common vs. locally-controlled TRI-STATE control. Additionally, the OLMCs select between alternate logic array input paths to maintain JEDEC cell-map compatibility with either "small-PAL" or "medium-PAL" logic arrays.

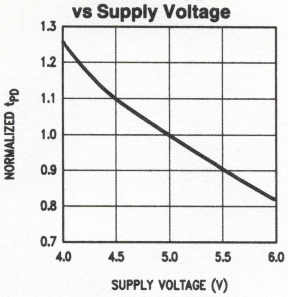
The various configurations of the OLMCs are controlled by a set of programmable "architecture" cells, separate from the logic-defining array cells. Each GAL device contains two "global" architecture cells, "SYN" and "AC0", which affect all OLMCs. Each of the devices's eight OLMCs also contains two "local" cells, "AC1" and "XOR". The OLMC Logic Diagram in *Figure 6* shows how the architecture cells select the different paths through the OLMC.

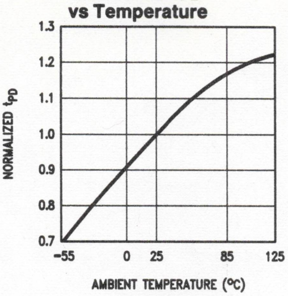
The SYN bit controls whether a device will have any registered outputs (SYN = 0) or will be purely combinatorial (SYN = 1). The SYN bit determines whether device pins* 1 and 13 are used as the clock and global TRI-STATE control inputs (SYN = 0) or whether they are ordinary inputs (SYN = 1). The AC0 bit selects between the "Small-PAL" mode and the "Medium/Registered-PAL" modes. The function of the AC1 bits depend on the state of the AC0 bit. In "Small-PAL" mode (AC0 = 0), the AC1 bit in each OLMC determines whether the associated device pin is an output (AC1 = 0) or an input (AC1 = 1). In "Registered-PAL" mode (AC0 = 1), the AC1 bit determines whether each OLMC is registered (AC1 = 0) or combinatorial (AC1 = 1). In "Medium-PAL" mode (AC0 = 1), the AC1 bits in all OLMCs must be set to 1 (combinatorial). All of the valid architecture bit configurations are shown in the OLMC Architecture table (Table IV), which has the same familiar format used in the OLMC Selection table (Table I).

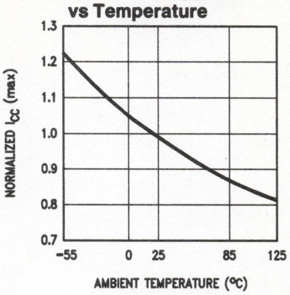
Independent of SYN, AC0 and the AC1 bits, the XOR bit in each OLMC selects between active-low (XOR = 0) or active-high (XOR = 1) output polarity.

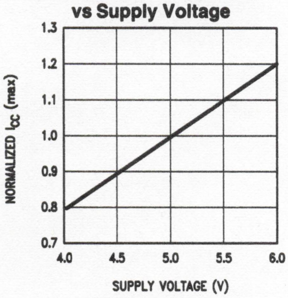
* Applies to 24-pin DIP packages for GAL20V8; refer to the 28-lead PCC Connection Diagram for conversion.

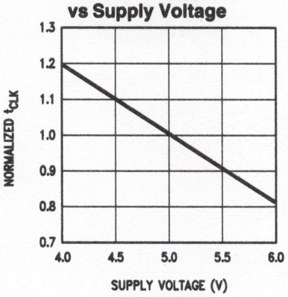
Typical Performance Characteristics

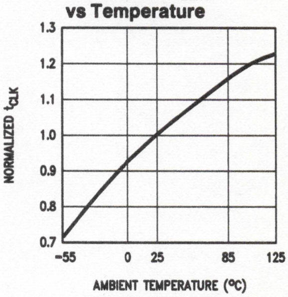

Normalized t_{PD} vs Supply Voltage

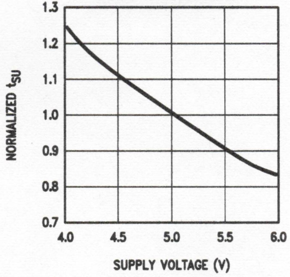

Normalized t_{PD} vs Temperature

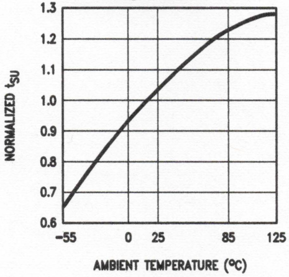

Normalized I_{CC} Max vs Temperature

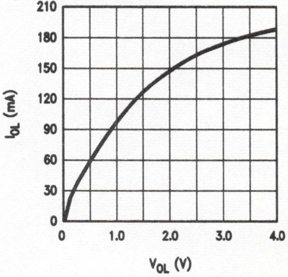

Normalized I_{CC} Max vs Supply Voltage

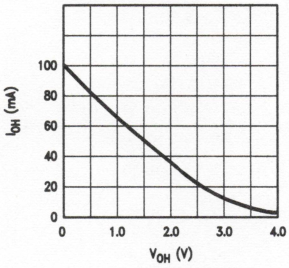

Normalized t_{CLK} vs Supply Voltage


Normalized t_{CLK} vs Temperature


Normalized t_{SU} vs Supply Voltage


Normalized t_{SU} vs Temperature


I_{OL} vs V_{OL}


I_{OH} vs V_{OH}

TL/L/9343–19

National Semiconductor

PRELIMINARY

GAL16V8A Generic Array Logic

General Description

The NSC E^{2}CMOS™ GAL® device combines a high performance CMOS process with electrically erasable floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels.

The 20-pin GAL16V8A features 8 programmable Output Logic Macrocells (OLMCs) allowing each output to be configured by the user. Additionally, the GAL16V8A is capable of emulating, in a functional/fuse map/parametric compatible device, all common 20-pin PAL® device architectures.

Programming is accomplished using readily available hardware and software tools. NSC guarantees a minimum 100 erase/write cycles.

Unique test circuitry and reprogrammable cells allow complete AC, DC, cell and functionality testing during manufacture. Therefore, NSC guarantees 100% field programmability and functionality of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.

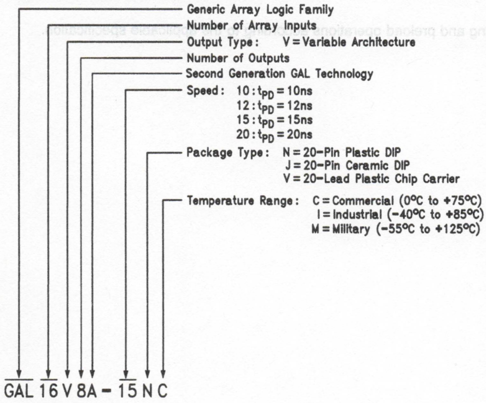
Features

- High performance E^{2}CMOS technology
  - — 10 ns maximum propagation delay
  - — 8 ns maximum from clock input to data output
  - — TTL compatible 24 mA outputs
  - — UltraMOS® III advanced CMOS technology
- 50% reduction in power
  - — 90 mA typ I_{CC}
- Electrically erasable cell technology
  - — Reconfigurable logic
  - — Reprogrammable cells
  - — 100% tested/guaranteed 100% yields
  - — High speed electrical erasure (<50 ms)
  - — 20 year data retention
- Eight output logic macrocells
  - — Maximum flexibility for complex logic designs
  - — Programmable output polarity
  - — Also emulates 20-pin PAL devices with full function/fuse map/parametric compatibility
- Preload and power-up reset of all registers
  - — 100% functional testability
- Applications include:
  - — DMA control
  - — State machine control
  - — High speed graphics processing
  - — Standard logic speed upgrade
- Electronic signature for identification
- Same JEDEC map as GAL16V8

PAL Replacement by Device Type

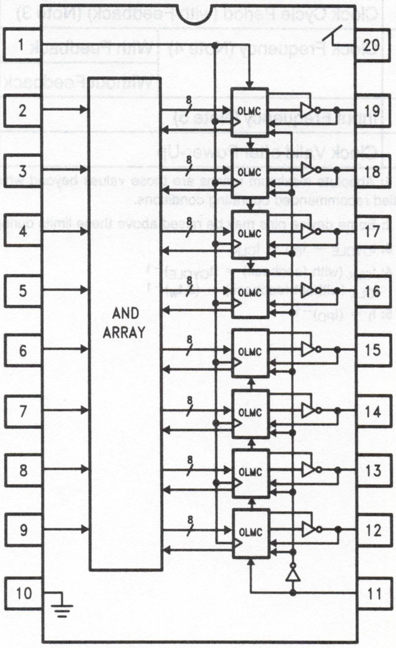
"Small PAL" Mode				"Registered PAL" Mode			"Medium PAL" Mode
10L8	12L6	14L4	16L2	16R8	16R6	16R4	16L8
10H8	12H6	14H4	16H2	16RP8	16RP6	16RP4	16H8
10P8	12P6	14P4	16P2				16P8

Ordering Information


Generic Array Logic Family
Number of Array Inputs
Output Type: V = Variable Architecture
Number of Outputs
Second Generation GAL Technology
Speed: 10: t_{PD} = 10ns
12: t_{PD} = 12ns
15: t_{PD} = 15ns
20: t_{PD} = 20ns
Package Type: N = 20-Pin Plastic DIP
J = 20-Pin Ceramic DIP
V = 20-Lead Plastic Chip Carrier
Temperature Range: C = Commercial (0°C to +75°C)
I = Industrial (−40°C to +85°C)
M = Military (−55°C to +125°C)
GAL 16 V 8A − 15 N C

TL/L/9999–1

Block Diagram—GAL16V8A


1
20
2
19
3
18
4
17
5
16
6
15
7
14
8
13
9
12
10
11
AND ARRAY
OLMC
8

TL/L/9999–2

Absolute Maximum Ratings (Note 1)

PRELIMINARY

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage V_{CC} (Note 2) −0.5V to +7.0V
Input Voltage (Note 2) −2.5V to V_{CC} +1.0V
Off-State Output Voltage (Note 2) −2.5V to V_{CC} +1.0V
Output Current ±100 mA
Storage Temperature −65°C to +150°C
Ambient Temperature with Power Applied −65°C to +125°C
Junction Temperature −65°C to +150°C
Lead Temperature (Soldering, 10 seconds) 260°C
ESD Tolerance TBD
C_{ZAP} = 100 pF
R_{ZAP} = 150Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-026

Recommended Operating Conditions

SUPPLY VOLTAGE AND TEMPERATURE

Symbol	Parameter	Commercial			Industrial			Military			Units
		Min	Nom	Max	Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	4.5	5	5.5	4.5	5	5.5	V
T_A	Operating Free-Air Temperature	0	25	75	−40	25	85	−55	25		°C
T_C	Operating Case Temperature									125	°C

AC TIMING REQUIREMENTS

Symbol	Parameter		GAL16V8A-10 COM		GAL16V8A-12 COM		GAL16V8A-15 COM IND/MIL		GAL16V8A-20 IND/MIL		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{SU}	Set-Up Time (Input or Feedback before Clock)		10		12		12		15		ns
t_H	Hold Time (Input after Clock)		0		0		0		0		ns
t_W	Clock Pulse Width (High/Low)		8		8		10		12		ns
t_{CYCLE}	Clock Cycle Period (with Feedback) (Note 3)		18		22		24		30		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		55.5		45.5		41.6		33.3	MHz
		Without Feedback		62.5		62.5		50.0		41.6	MHz
f_I	Input Frequency (Note 5)			100.0		83.3		66.6		50.0	MHz
t_{PR}	Clock Valid after Power-Up			100		100		100		100	ns

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: $t_{CYCLE} = t_{SU} + t_{CLK}$

Note 4: f_{CLK} (with feedback) $= (t_{CYCLE})^{-1}$
f_{CLK} (without feedback) $= (2\ t_W)^{-1}$

Note 5: $f_I = (t_{PD})^{-1}$

Electrical Characteristics

Over Recommended Operating Conditions

PRELIMINARY

Symbol	Parameter	Conditions		Temperature Range	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage				2.0		$V_{CC}+1$	V
V_{IL}	Low Level Input Voltage				−1.0		0.8	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −3.2 mA	COM/IND	2.4			V
			I_{OH} = −2.0 mA	MIL	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 24 mA	COM/IND			0.5	V
			I_{OL} = 12 mA	MIL			0.5	V
I_{OZH}	High Level Off State Output Current	V_{CC} = Max, V_O = V_{CC}(Max)					10	μA
I_{OZL}	Low Level Off State Output Current	V_{CC} = Max, V_O = GND					−10	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = V_{CC}(Max)					10	μA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = V_{CC}(Max)					10	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = GND					−10	μA
I_{OS}*	Output Short Circuit Current	V_{CC} = 5.0V, V_O = GND			−30		−160	mA
I_{CC}	Supply Current	f = 15 MHz, V_{CC} = Max		COM			115	mA
				MIL/IND			140	mA
C_I	Input Capacitance	V_{CC} = 5.0V, V_I = 2.0V					12	pF
$C_{I/O}$	I/O Capacitance	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V					15	pF

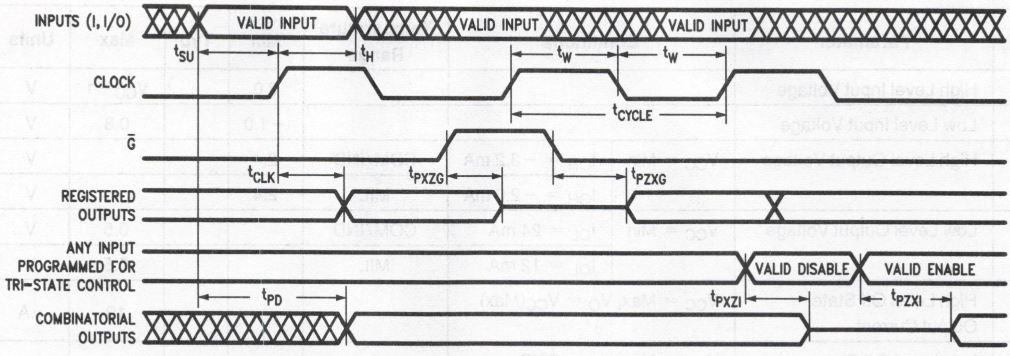
*One output at a time for a maximum duration of one second.

Switching Characteristics

Over Recommended Operating Conditions

Symbol	Parameter	Conditions	GAL16V8A-10		GAL16V8A-12		GAL16V8A-15		GAL16V8A-20		Units
			COM		COM		COM IND/MIL		IND/MIL		
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{PD}	Input or Feedback to Combinatorial Output	S1 Closed, C_L = 50 pF		10		12		15		20	ns
t_{CLK}	Clock to Registered Output or Feedback	S1 Closed, C_L = 50 pF		8		10		12		15	ns
t_{PZXG}	$\overline{G}$ ↓ to Registered Output Enabled	Active High: S1 Open, C_L = 50 pF Active Low: S1 Closed, C_L = 50 pF		10		10		15		18	ns
t_{PXZG}	$\overline{G}$ ↑ to Registered Output Disabled	From V_{OH}: S1 Open, C_L = 5 pF From V_{OL}: S1 Closed, C_L = 5 pF		10		10		15		18	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	Active High: S1 Open, C_L = 50 pF Active Low: S1 Closed, C_L = 50 pF		10		12		15		20	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	From V_{OH}: S1 Open, C_L = 5 pF From V_{OL}: S1 Closed, C_L = 5 pF		10		12		15		20	ns
t_{RESET}	Power-Up to Registered Output High	S1 Closed, C_L = 50 pF		45		45		45		45	μs

Switching Waveforms


INPUTS (I, I/O)
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
t_{CYCLE}
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/9999-9

National Semiconductor

PRELIMINARY

GAL20V8A
Generic Array Logic

General Description

The NSC E²CMOS™ GAL® device combines a high performance CMOS process with electrically erasable floating gate technology. This programmable memory technology applied to array logic provides designers with reconfigurable logic and bipolar performance at significantly reduced power levels.

The 24-pin GAL20V8A features 8 programmable Output Logic Macrocells (OLMCs) allowing each output to be configured by the user. Additionally, the GAL20V8A is capable of emulating, in a functional/fuse map/parametric compatible device, the most popular 24-pin PAL® device architectures.

Programming is accomplished using readily available hardware and software tools. NSC guarantees a minimum 100 erase/write cycles.

Unique test circuitry and reprogrammable cells allow complete AC, DC, cell and functionality testing during manufacture. Therefore, NSC guarantees 100% field programmability of the GAL devices. In addition, electronic signature is available to provide positive device ID. A security circuit is built-in, providing proprietary designs with copy protection.

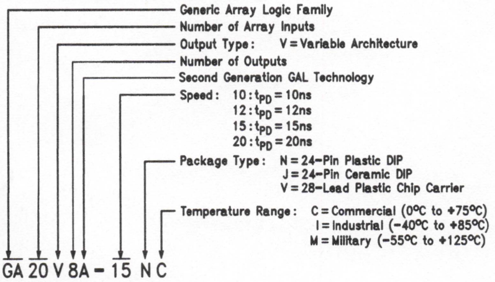
Features

- High performance E²CMOS technology
  - — 10 ns maximum propagation delay
  - — f_{max} = 62.5 MHz
  - — 8 ns maximum from clock input to data output
  - — TTL compatible 24 mA outputs
  - — UltraMOS® III advanced CMOS technology
- 50% reduction in power
  - — 90 mA typ I_{CC}
- Electrically erasable cell technology
  - — Reconfigurable logic
  - — Reprogrammable cells
  - — 100% tested/guaranteed 100% yields
  - — High speed electrical erasure (<50 ms)
  - — 20 year data retention
- Eight output logic macrocells
  - — Maximum flexibility for complex logic designs
  - — Programmable output polarity
  - — Also emulates 24-pin PAL devices with full function/fuse map/parametric compatibility
- Preload and power-up reset of all registers
  - — 100% functional testability
- Applications include:
  - — DMA control
  - — State machine control
  - — High speed video graphics processing
  - — Standard logic speed upgrade
- Electronic signature for identification
- Same JEDEC map as GAL20V8

PAL Replacement by Device Type

"Small PAL" Mode				"Registered PAL" Mode			"Medium PAL" Mode
14L8	16L6	18L4	20L2	20R8	20R6	20R4	20L8
14H8	16H6	18H4	20H2	20RP8	20RP6	20RP4	20H8
14P8	16P6	18P4	20P2				20P8

Ordering Information

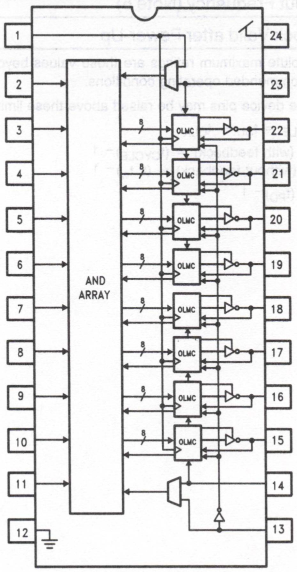


Generic Array Logic Family
Number of Array Inputs
Output Type: V = Variable Architecture
Number of Outputs
Second Generation GAL Technology
Speed: 10 : t_{PD} = 10ns
12 : t_{PD} = 12ns
15 : t_{PD} = 15ns
20 : t_{PD} = 20ns
Package Type: N = 24-Pin Plastic DIP
J = 24-Pin Ceramic DIP
V = 28-Lead Plastic Chip Carrier
Temperature Range: C = Commercial (0°C to +75°C)
I = Industrial (−40°C to +85°C)
M = Military (−55°C to +125°C)

GA 20 V 8A − 15 N C

TL/L/10000–2

Block Diagram—GAL20V8A



TL/L/10000–1

Absolute Maximum Ratings (Note 1)

PRELIMINARY

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 2)	−0.5V to +7.0V
Input Voltage (Note 2)	−2.5V to V_{CC} +1.0V
Off-State Output Voltage (Note 2)	−2.5V to V_{CC} +1.0V
Output Current	±100 mA
Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−65°C to +125°C
Junction Temperature	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	260°C
ESD Tolerance	TBD

C_{ZAP} = 100 pF
R_{ZAP} = 150Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

SUPPLY VOLTAGE AND TEMPERATURE

Symbol	Parameter	Commercial			Industrial			Military			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage	4.75	5	5.25	4.5	5	5.5	4.5	5	5.5	V
T_A	Operating Free-Air Temperature	0	25	75	−40	25	85	−55	25		°C
T_C	Operating Case Temperature									125	°C

AC TIMING REQUIREMENTS

Symbol	Parameter		GAL20V8A-10 COM		GAL20V8A-12 COM		GAL20V8A-15 COM IND/MIL		GAL20V8A-20 IND/MIL		Units
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{SU}	Set-Up Time (Input or Feedback before Clock)		10		12		12		15		ns
t_H	Hold Time (Input after Clock)		0		0		0		0		ns
t_W	Clock Pulse Width (High/Low)		8		8		10		12		ns
t_{CYCLE}	Clock Cycle Period (with Feedback) (Note 3)		18		22		24		30		ns
f_{CLK}	Clock Frequency (Note 4)	With Feedback		55.5		45.5		41.6		33.3	
		Without Feedback		62.5		62.5		50.0		41.6	MHz
f_I	Input Frequency (Note 5)			100.0		83.3		66.6		50.0	
t_{PR}	Clock Valid after Power-Up			100		100		100		100	ns

Note 1: Absolute maximum ratings are those values beyond which the device may be permanently damaged. Proper operation is not guaranteed outside the specified recommended operating conditions.

Note 2: Some device pins may be raised above these limits during programming and preload operations according to the applicable specification.

Note 3: $t_{CYCLE} = t_{SU} + t_{CLK}$

Note 4: f_{CLK} (with feedback) $= (t_{CYCLE})^{-1}$
f_{CLK} (without feedback) $= (2\ t_W)^{-1}$

Note 5: $f_I = (t_{PD})^{-1}$

Electrical Characteristics

Over Recommended Operating Conditions

PRELIMINARY

Symbol	Parameter	Conditions		Temperature Range	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage				2.0		$V_{CC}+1$	V
V_{IL}	Low Level Input Voltage				−1.0		0.8	V
V_{OH}	High Level Output Voltage	V_{CC} = Min	I_{OH} = −3.2 mA	COM/IND	2.4			V
			I_{OH} = −2.0 mA	MIL	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min	I_{OL} = 24 mA	COM/IND			0.5	V
			I_{OL} = 12 mA	MIL			0.5	V
I_{OZH}	High Level Off State Output Current	V_{CC} = Max, V_O = V_{CC} (Max)					10	µA
I_{OZL}	Low Level Off State Output Current	V_{CC} = Max, V_O = GND					−10	µA
I_I	Maximum Input Current	V_{CC} = Max, V_I = V_{CC} (Max)					10	µA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = V_{CC} (Max)					10	µA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = GND					−10	µA
I_{OS}*	Output Short Circuit Current	V_{CC} = 5.0V, V_O = GND			−30		−160	mA
I_{CC}	Supply Current	f = 15 MHz, V_{CC} = Max		COM			115	mA
				MIL/IND			140	mA
C_I	Input Capacitance	V_{CC} = 5.0V, V_I = 2.0V					12	pF
$C_{I/O}$	I/O Capacitance	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V					15	pF

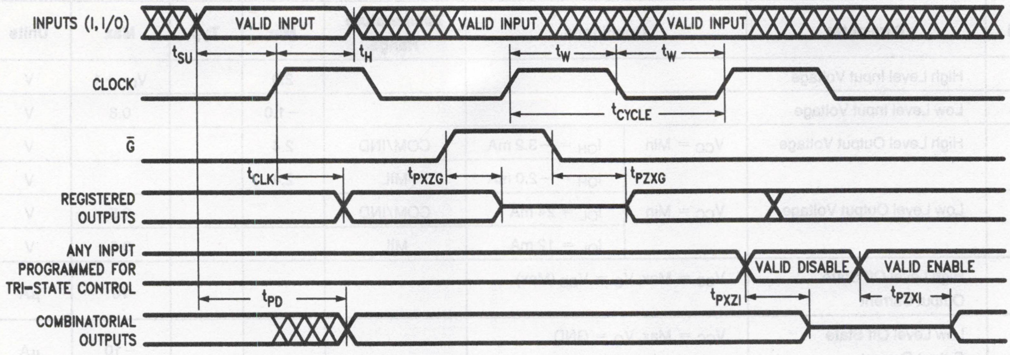
*One output at a time for a maximum duration of one second.

Switching Characteristics

Over Recommended Operating Conditions

Symbol	Parameter	Conditions	GAL20V8A-10		GAL20V8A-12		GAL20V8A-15		GAL20V8A-20		Units
			COM		COM		COM IND/MIL		IND/MIL		
			Min	Max	Min	Max	Min	Max	Min	Max	
t_{PD}	Input or Feedback to Combinatorial Output	S1 Closed, C_L = 50 pF		10		12		15		20	ns
t_{CLK}	Clock to Registered Output or Feedback	S1 Closed, C_L = 50 pF		8		10		12		15	ns
t_{PZXG}	$\overline{G}$ ↓ to Registered Output Enabled	Active High; S1 Open, C_L = 50 pF Active Low; S1 Closed, C_L = 50 pF		10		10		15		18	ns
t_{PXZG}	$\overline{G}$ ↑ to Registered Output Disabled	From V_{OH}; S1 Open, C_L = 5 pF From V_{OL}; S1 Closed, C_L = 5 pF		10		10		15		18	ns
t_{PZXI}	Input to Combinatorial Output Enabled via Product Term	Active High; S1 Open, C_L = 50 pF Active Low; S1 Closed, C_L = 50 pF		10		12		15		20	ns
t_{PXZI}	Input to Combinatorial Output Disabled via Product Term	From V_{OH}; S1 Open, C_L = 5 pF From V_{OL}; S1 Closed, C_L = 5 pF		10		12		15		20	ns
t_{RESET}	Power-Up to Registered Output High	S1 Closed, C_L = 50 pF		45		45		45		45	µs

Switching Waveforms


INPUTS (I, I/O)
VALID INPUT
VALID INPUT
VALID INPUT
t_{SU}
t_H
t_W
t_W
CLOCK
t_{CYCLE}
$\overline{G}$
t_{CLK}
t_{PXZG}
t_{PZXG}
REGISTERED OUTPUTS
ANY INPUT PROGRAMMED FOR TRI-STATE CONTROL
VALID DISABLE
VALID ENABLE
t_{PD}
t_{PXZI}
t_{PZXI}
COMBINATORIAL OUTPUTS

TL/L/10000-9

National Semiconductor

ADVANCE INFORMATION

GAL16Z8 In-System re-Programmable (isp) Generic Array Logic

General Description

The NSC GAL®16Z8 is a revolutionary programmable logic device featuring 5V only in-system reprogrammability and real time, in-system diagnostic capabilties. This is made possible by on-chip circuitry which generates and shapes the necessary high voltage internal programming control signals. Using NSC UltraMOS® technology, this device provides true bipolar performance at significantly reduced power levels.

The 24-pin GAL16Z8 is architecturally and parametrically identical to the 20-pin GAL16V8, but includes 4 extra pins to control in-system programming. These extra pins are: data clock (DCLK), serial data in (SDI), serial data out (SDO), and mode control (MODE). These pins are not associated with normal logic functions and are used only during programming. Additionally, this 4-pin interface allows an unlimited number of devices to be cascaded to form a serial programming and diagnostics loop.

Advanced features that simplify programming and reduce test time, coupled with E²CMOS™ reprogrammable cells, enable complete AC, DC, programmability, and functionality testing of each GAL16Z8 during manufacturing and allows National Semiconductor to guarantee 100% performance to all specifications.

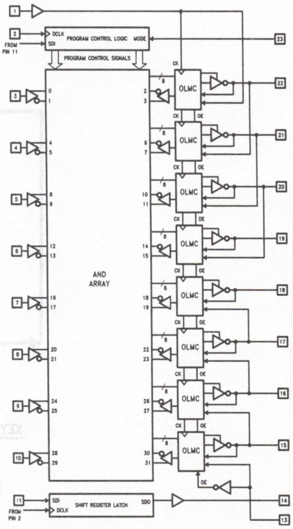
Features

- In-system reconfigurable—5V only programming
  - — Change logic "On the Fly" (in less than 1s)
  - Nonvolatile electrically erasable technology
- Diagnositics mode for controllability and observability of system logic
- High performance E²CMOS technology
  - High speed: 25 ns max propagation delay
  - Low power: 90 mA max active
- Eight output logic macrocells
  - — Maximum flexibility for complex logic designs
  - — Also emulates 20-pin PAL® devices with full function/fuse map/parametric compatibility
- Preload and power-up reset of all registers
  - — 100% functional testability
- Space saving 24-pin, 300-mil DIP
- Minimum 10,000 erase/write cycles
- Data retention exceeds 20 years
- Electronic signature for identification
- Applications include:
  - — Reconfigurable interfaces
  - — Copy protection and security schemes
  - — erasable hardware
  - — password systems
  - — proprietary hardware/software interlocks

PAL Replacement by Device Type

"Small-PAL" Mode				"Registered-PAL" Mode			"Medium-PAL" Mode
10L8 10H8 10P8	12L6 12H6 12P6	14L4 14H4 14P4	16L2 16H2 16P2	16R8 16RP8	16R6 16RP6	16R4 16RP4	16L8 16H8 16P8

Block Diagram—GAL16Z8



TL/L/10001-1

Overview

The GAL16Z8 device has three basic modes of operation: NORMAL, DIAGNOSTIC and PROGRAM. These three modes are controlled by the system designer through the use of a sophisticated on-chip state machine.

In addition, the GAL16Z8 has been optimized so that the use of 2 or more devices on a board requires the same amount of control overhead as a single device would. This Serial Loop approach applies to the DIAGNOSTIC and PROGRAM operation modes of the device.

The balance of this document will perform a general review of the operation of the GAL16Z8 in its various modes and will explain the use of these control pins.

Mode Control and Operation

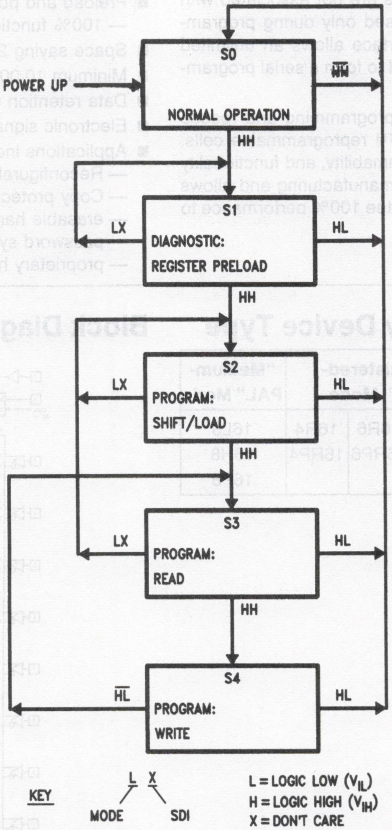
The signals used to control this device are the same for both the DIAGNOSTIC and PROGRAM modes. During NORMAL mode, the control pins serve no function other than to control the transition to another mode. The shared control pin approach allows for a simple multi-mode operation with minimal system or board overhead.

The four control signals are TTL level signals; MODE, DCLK, SDI and SDO. These signals are used to transition from mode to mode and through each of the five states as shown in *Figure 1*. MODE is used only to control the on-chip state machine. DCLK is used for mode control and for the orderly clocking of data into the 16Z8 from the SDI (serial data in) pin as well as out of the device through the SDO (serial data out) pin. SDI is also used for state machine control.

The current state cannot be explicitly observed, however, an "escape" sequence ("HL") to the NORMAL mode is always available to start the process fresh. Caution should be used when using this escape path as the pattern in the device may not be valid if an erase or reprogramming operation was in progress.

Diagnostic Mode

From the NORMAL mode, the device transitions to the DIAGNOSTIC:Preload state (mode:state). In this mode, the values in the Macrocell registers can be interrogated or "pre-"loaded for diagnostic testing of the system. Advanced system design requires full control and observability of all registers on a board.



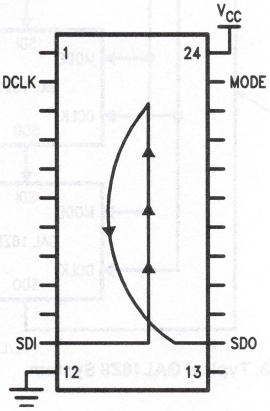
TL/L/10001-3

FIGURE 1. On-Chip State Machine

Diagnostic Mode (Continued)

Upon entry to the DIAGNOSTIC:Preload state, the data on the device output pins (15-22) is latched and held to its 1, 0 or TRI-STATE® condition. This is important as the DIAGNOSTIC:Preload state configures a serial loop from the SDI pin through each of the registers to the SDO pin. Data is shifted across all of the registers during diagnostics *(Figure 2)*. The latching of the current output data insures that the system is not influenced by the changing register contents until such time as the DIAGNOSTIC:Preload state is exited.

The access to the macrocell data is through the SDI and SDO pins. While in the DIAGNOSTIC:Preload state, the value in each register is serially shifted out through SDO with each pulse of the DCLK pin. Similarly, new data can be preloaded into the registers through the SDI pin.



TL/L/10001-4

FIGURE 2. Diagnostic/Program Pinout

The number of registers in a 16Z8 is a function of the configuration of each macrocell. The length of the serial path, and therefore the number of bits of data shifted in or out of the device, is a function of the number of macrocells which are configured to have registers.

Program Mode

The PROGRAM mode can only be entered from the DIAGNOSTIC:Preload state. When this transition is made, the value of the user programmable TRI-STATE Bit (TSB) is examined to determine the data condition that is held on the device output pins 15–22. The data can either remain latched to 1-0-Z as in the DIAGNOSTIC mode or the data can be forced to high impedance. Again, this feature allows complete control of the system during programming.

The PROGRAM mode consists of three states of the on-chip state machine: SHIFT, READ and WRITE. Proper sequencing of these states is necessary to program and verify the device. Programming and verification is accomplished using a Serial Register Latch (SRL) to program or verify a row of data at a time.

PROGRAM:SHIFT

During the PROGRAM:Shift state, the 88-bit SRL is serially loaded with 82 bits of data and 6 bits of row address for each row to be programmed. The architecture and Electronic Signature of the 16Z8 are also programmed in the same manner. DCLK is used to shift data into SDI for the loading process.

PROGRAM:READ

Verification of data in the array is accomplished in the PROGRAM:Read state. Exiting the PROGRAM:Read state to the PROGRAM:Shift state causes the contents of the array row to be copied to the SRL. This data can be shifted out as outlined above. Programming the Security Cell prevents valid data from being loaded into the SRL. This feature is provided to prevent subsequent copying of the cell patterns.

PROGRAM:WRITE

The actual programming cycle occurs in the PROGRAM:Write state. The data to be programmed is loaded into the SRL in the PROGRAM:Shift state prior to executing the write cycle. It is the responsibility of the system control logic to assure that the device stays in the PROGRAM:Write state for a sufficient time to program the E^2 cells, approximately 10 ms. The PROGRAM:Write is then exited to the PROGRAM:Read state for verification of the data.

The 16Z8 is completely erased by addressing an "erase" row address using the same process outlined above. It is necessary to bulk-erase the device prior to rewriting any pattern into the device as each row write cycle does not include an automatic erase of that row.

The entire programming process takes less than ½ second. The bulk erase is 10 ms, and each of the 36 programmable rows can be loaded, programmed and verified in approximately 10.5 ms for a total time of 0.39 seconds. During this time, the device output pins are latched or in the high impedance condition and the 16Z8 is not responding to changes on its input pins. The system must accommodate this programming time.

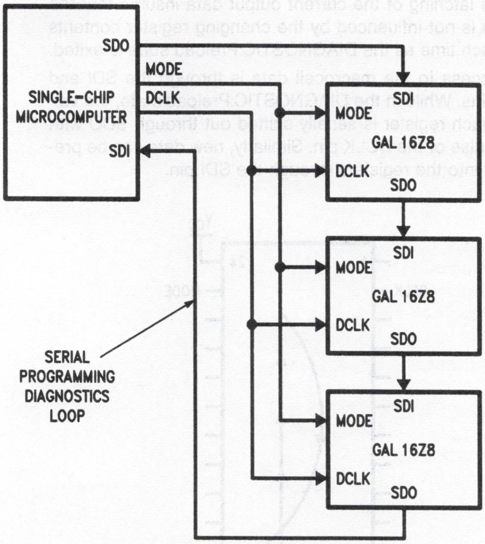
Serial Diagnostic/Program Loop Operation

Figure 3 shows a typical GAL16Z8 system. Notice that several devices have been cascaded together to form a serial programming loop. This arrangement allows the simultaneous transfer of programming and diagnostic data through every In-System re-Programmable GAL device in the system with no additional control logic necessary. When controlling multiple devices in such a loop, the basic diagnostic and programming algorithms remain unchanged. However, there are some additional considerations.

In a serial programming loop, the SDO of the first device is connected to the SDI of the second; the SDO of the second to the SDI of the third, and so on. DCLK and MODE are common for every device. With such an arrangement, devices in the loop are always in the same state, but the data being shifted into their respective SRLs may be different. Note that, before data reaches the SDI input of any given device, it must first pass through the 88-bit SRL of every device ahead of it in the loop. The SRL is asynchronously bypassed (SDO = SDI) whenever MODE = 1 allowing SDI to function as both a data and mode control pin.

In a serial diagnostic loop, the length of the loop is a function of the number of OLMC registers being used. On the other hand, in a serial programming loop, data transfers always occur in multiples of 88 bits, as the data must pass through the SRL of other devices in the loop.

Because all devices in a loop are always in the same state, reprogramming just one out of "n" devices would seem to be a problem. This, however, is not the case, as several options exist. The most obvious solution is to simply reprogram all of the devices, even if the new pattern is the same as the old. The system "down" time is effectively the same since all the devices reprogram in parallel.



TL/L/10001-5

FIGURE 3. Typical GAL16Z8 System

National Semiconductor

ADVANCE INFORMATION

GAL39V18 Generic Array Logic

General Description

Using a high performance E²CMOS™ technology, National Semiconductor has produced a next-generation programmable logic device, the GAL®39V18. Having an FPLA architecture, known for its superior flexibility in state-machine design, the GAL39V18 offers the highest degree of functional integration, flexibility, and speed currently available in a 24-pin, 300-mil package.

The GAL39V18 has 10 programmable Output Logic MacroCells (OLMC) and 8 programmable "buried" State Logic MacroCells (SLMC). In addition, there are 10 Input Logic MacroCells (ILMC) and 10 I/O Logic MacroCells (IOLMC). Two clock inputs are provided for independent control of the input and output macrocells.

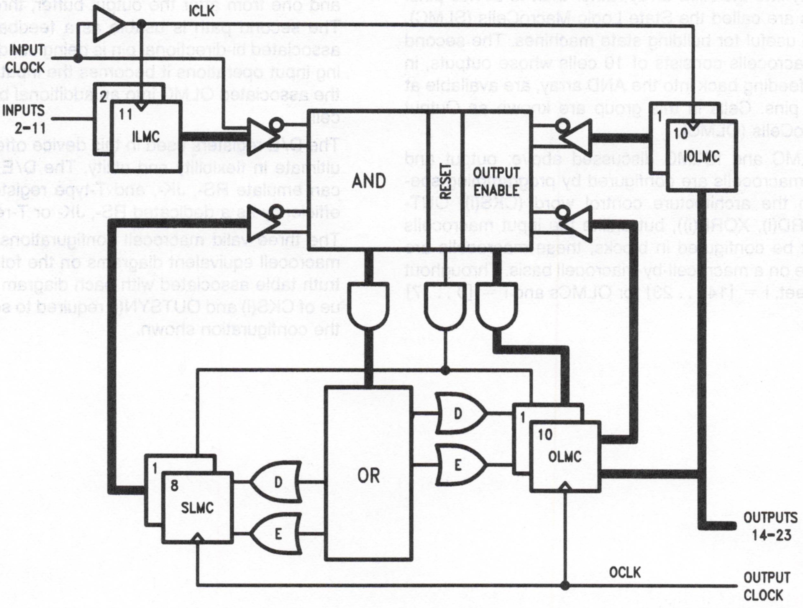
Advanced features that simplify programming and reduce test time, coupled with E²CMOS reprogrammable cells, enable 100% AC, DC, programmability, and functionality testing of each GAL39V18 during manufacture. This allows National to guarantee 100% performance to specifications. In addition, data retention of 20 years and a minimum of 100 erase/write cycles are guaranteed.

Programming is accomplished using standard hardware and software tools. In addition, an Electronic Signature word is available for storage of user specified data, and a security cell is provided to protect proprietary designs.

Features

- Electrically erasable cell technology
  - — Instantly reconfigurable logic
  - — Instantly reprogrammable cells
  - — Guaranteed 100% yields
- High performance E²CMOS technology
  - — Low power: 90 mA Typical
  - — High speed:
    - 15 ns max. clock to output delay
    - 25 ns max. setup time
    - 30 ns max. propagation delay
- TTL compatible inputs and outputs
- Unprecedented functional density
  - — 10 output logic macrocells
  - — 8 state logic macrocells
  - — 20 input and I/O logic macrocells
- High-level design flexibility
  - — 78 × 64 × 36 FPLA architecture
  - — Separate state register and input clock pins
  - — Functionally supersets existing 24-pin PAL® and IFL™ devices
  - — Asynchronous clocking
- Space saving 24-pin, 300-mil DIP
- High speed programming algorithm
- 20-year data retention

Block Diagram—GAL39V18



TL/L/10002-1

Input Logic MacroCell (ILMC) and I/O Logic MacroCell (IOLMC)

The GAL39V18 features two configurable input sections. The ILMC section corresponds to the dedicated input pins (2–11) and the IOLMC to the I/O pins (14–23). Each input section is configurable as a block for asynchronous, latched, or registered inputs. Pin 1 (ICLK) is used as an enable input for latched macrocells (transparent when high) and as a clock for registered macrocells (positive edge triggered).

Configurable input blocks can be used to advantage by system designers. Registered inputs are popular for synchronization and data merging. Transparent latches are useful when the input data is invalid outside a known time window. Direct inputs are used in systems where the input data is well ordered in time. With the GAL39V18, external registers and latches are not necessary.

The various configurations of the input and I/O macrocells are controlled by programming four architecture control bits (INLATCH, INSYN, IOLATCH, IOSYN) within the 68-bit architecture control word. The SYN bits determine whether the macrocells will have register/latch capability or will be strictly asynchronous. The LATCH bits select between latched and registered inputs.

The three valid macrocell configurations are shown in the macrocell equivalent diagrams on the following pages. The truth table associated with each diagram shows the values of the LATCH and SYN bits requried to set the macrocell to the configuration shown.

Output Logic MacroCell (OLMC)/ State Logic MacroCell (SLMC)

The outputs of the OR array feed two groups of macrocells. One group of eight macrocells is buried; its outputs feed back directly into the AND array rather than to device pins. These cells are called the State Logic MacroCells (SLMC), as they are useful for building state machines. The second group of macrocells consists of 10 cells whose outputs, in addition to feeding back into the AND array, are available at the device pins. Cells in this group are known as Output Logic MacroCells (OLMC).

Like the ILMC and IOLMC discussed above, output and state logic macrocells are configured by programming specific bits in the architecture control word (CKS(i), OUTSYN(i), XORD(i), XORE(i)), but unlike the input macrocells which must be configured in blocks, these macrocells are configurable on a macrocell-by-macrocell basis. Throughout this datasheet, i = [14 . . . 23] for OLMCs and i = [0 . . . 7] for SLMCs.

State and Output Logic MacroCells may be set to one of three valid configurations: combinational, D-type registered with sum term (asynchronous) clock, or D/E-type registered. Output macrocells always have I/O capability, with directional control provided by the 10 output enable (OE) product terms. Additionally, the polarity of each OLMC output is selectable through the XORD(i) architecture bits. Polarity selection is not necessary for SLMCs, since both the true and complemented forms of their outputs are available in the AND array. Polarity of all "E" sum terms is selectable through the XORE(i) architecture control bits.

When CKS(i) = 1 and OUTSYN(i) = 0, macrocell "i" is set as "D/E-type registered." In this configuration, the register is clocked from the common OCLK and the register clock enable input is controlled by the associated "E" sum term. This configuration is useful for building counters and state-machines with state hold functions.

When CKS(i) = 0 and OUTSYN(i) = 0, macrocell "i" is set as "D-type registered with sum term clock." In this configuration, the register is enabled and its "E" sum term is routed directly to the clock input. This allows for the popular "asynchronous programmable clock" feature, selectable on a register-by-register basis.

When CKS(i) = 0 and OUTSYN(i) = 1, macrocell "i" is set as "combinatorial." Configuring a SLMC in this manner turns it into a complement array. Complement arrays are used to construct multi-level logic.

Registers in both the Output and State Logic MacroCells feature a RESET input. This active high input allows the registers to be simultaneously and asynchronously reset from a common signal. The source of this signal is the RESET product term. Registers reset to a logic zero, but since the output buffers invert, a logic one will be present at the device pins.

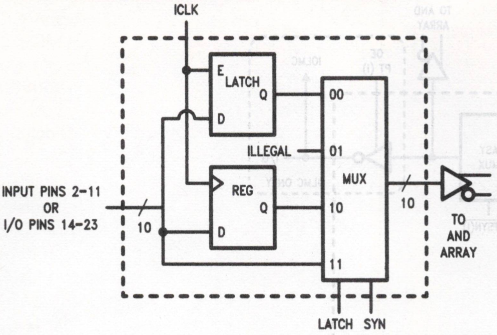
There are two possible feedback paths from each OLMC: one from before the output buffer (this is the normal path), and one from after the output buffer, through the IOLMCs. The second path is usable as a feedback only when the associated bi-directional pin is being used as an output; during input operations it becomes the input data path, turning the associated OLMC into an additional buried state macrocell.

The D/E registers used in this device offer the designer the ultimate in flexibility and utility. The D/E register construct can emulate RS-, JK-, and T-type registers with the same efficiency as a dedicated RS-, JK- or T-register.

The three valid macrocell configurations are shown in the macrocell equivalent diagrams on the following pages. The truth table associated with each diagram shows the bit value of CKS(i) and OUTSYN(i) required to set the macrocell to the configuration shown.

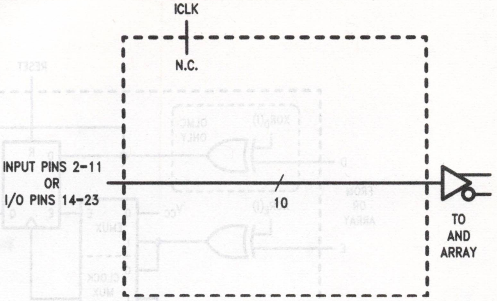
ILMC/IOLMC Configurations

ILMC/IOLMC Generic Block Diagram


ICLK
E
LATCH
Q
D
00
ILLEGAL
01
MUX
REG
Q
D
10
11
INPUT PINS 2-11
OR
I/O PINS 14-23
10
10
TO
AND
ARRAY
LATCH
SYN

TL/L/10002-4

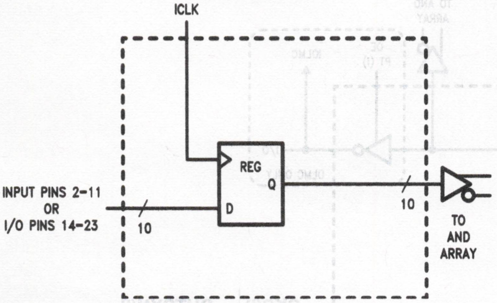
Asynchronous Input


ICLK
N.C.
INPUT PINS 2-11
OR
I/O PINS 14-23
10
TO
AND
ARRAY

TL/L/10002-5

LATCH	SYN
1	1

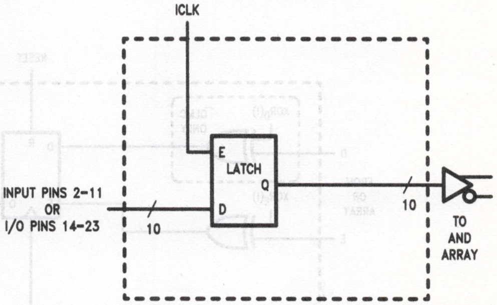
Registered Input


ICLK
REG
Q
D
INPUT PINS 2-11
OR
I/O PINS 14-23
10
10
TO
AND
ARRAY

TL/L/10002-6

LATCH	SYN
1	0

Latched Input

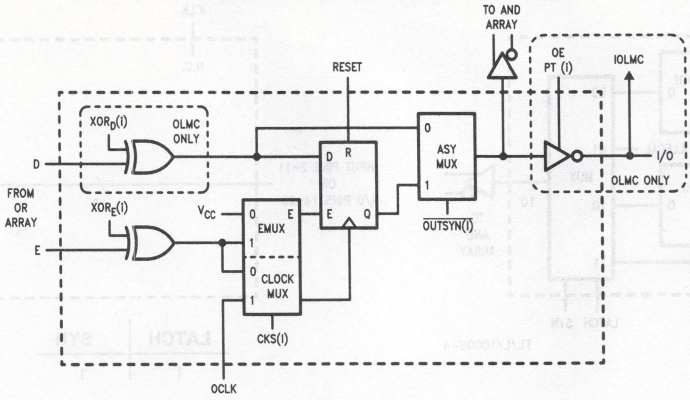

ICLK
E
LATCH
Q
D
INPUT PINS 2-11
OR
I/O PINS 14-23
10
10
TO
AND
ARRAY

TL/L/10002-7

LATCH	SYN
0	0

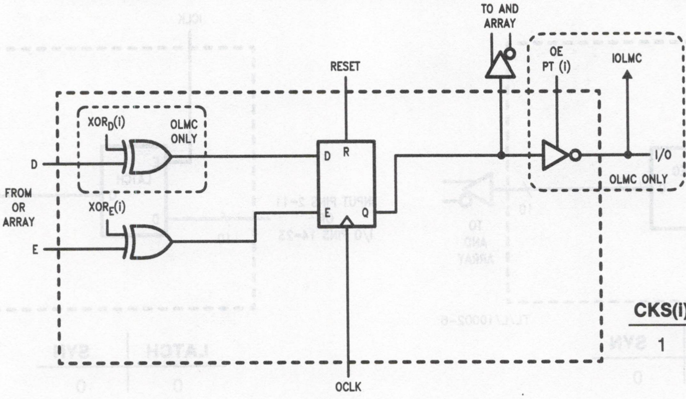
OLMC/SLMC Configurations

OLMC/SLMC Block Diagram


TO AND
ARRAY
RESET
OE
PT (i)
IOLMC
XOR$_D$(i)
OLMC
ONLY
D
FROM
OR
ARRAY
XOR$_E$(i)
E
V$_{CC}$
0
E
EMUX
1
0
CLOCK
MUX
1
D R
E Q
0
ASY
MUX
1
OUTSYN(i)
CKS(i)
I/O
OLMC ONLY
OCLK

TL/L/10002-8

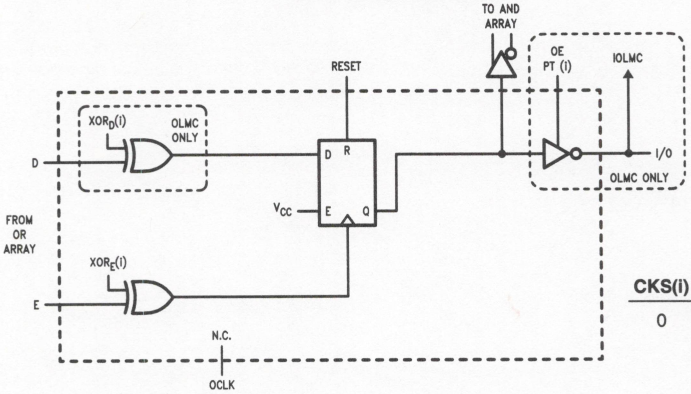
D/E Type Registered


TO AND
ARRAY
RESET
OE
PT (i)
IOLMC
XOR$_D$(i)
OLMC
ONLY
D
FROM
OR
ARRAY
XOR$_E$(i)
E
D R
E Q
I/O
OLMC ONLY
OCLK

CKS(i)	OUTSYN(i)
1	0

TL/L/10002-9

D-Type Registered with Sum Term Asynchronous Clock

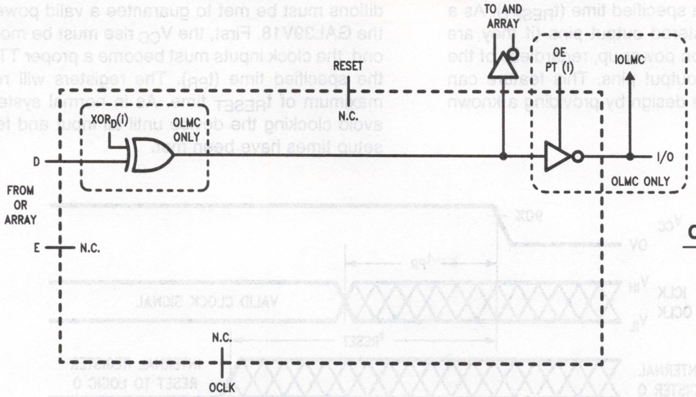

TO AND
ARRAY
RESET
OE
PT (i)
IOLMC
XOR$_D$(i)
OLMC
ONLY
D
V$_{CC}$
D R
E Q
I/O
OLMC ONLY
FROM
OR
ARRAY
XOR$_E$(i)
E
N.C.
OCLK

CKS(i)	OUTSYN(i)
0	0

TL/L/10002-10

OLMC/SLMC Configurations

Combinatorial



CKS(i)	OUTSYN(i)
0	1

TL/L/10002-11

Array Description

The GAL39V18 E^2 reprogrammable array is subdivided into three smaller arrays: AND, OR and Architecture. These arrays are described in detail below.

AND ARRAY

The AND array is organized as 78 input terms by 75 product term outputs. The 20 input and I/O logic macrocells, 8 SLMC feedbacks, 10 OLMC feedbacks, and ICLK comprise the 39 inputs to this array (each available in true and complemented forms). Product terms 0–63 serve as inputs to the OR array. Product term 64 is the RESET PT; it generates the RESET signal described in the earlier discussion of output and state logic macrocells. Product terms 65–74 are the output enable product terms; they control the output buffers, thus enabling device pins 14–23 to be bi-direction or TRI-STATE®.

OR ARRAY

The OR array is organized as 64 inputs by 36 sum term outputs. Product terms 0–63 of the AND array serve as the inputs to this array. Of the 36 sum term outputs, 18 are data ("D") terms and 18 are enable/clock ("E") terms. These terms feed into the 10 OLMCs and 8 SLMCs, one "D" term and one "E" term to each.

ARCHITECTURE ARRAY

The various configurations of the GAL39V18 are enabled by programming cells within the architecture control word. This 68-bit word contains all of the chip configuration data. This data includes: XORD(i), XORE(i), CKS(i), OUTSYN(i), INLATCH, INSYN, IOLATCH, and IOSYN. The function of each of these bits has been previously explained.

Electronic Signature Word

Every GAL39V18 device contains an electronic signature word. The Electronic Signature word is a 72-bit user definable storage area, which can be used to store inventory control data, pattern revision numbers, manufacture date, etc. Signature data is always available to the user, regardless of the state of the security cell.

Security Cell

A security cell is provided with every GAL39V18 device as a deterrent to unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the AND, OR and architecture arrays. This cell can be erased only during a bulk erase cycle, so the original configuration can never be examined once this cell is programmed. Electronic Signature data is always available to the user, regardless of the state of this control cell.

Bulk Erase

Before writing a new pattern into a previously programmed part, the old pattern must first be erased. This erasure is done automatically by the programming hardware as part of the programming cycle and takes only 50 ms.

Register Preload

When testing state machine designs, all possible states and state transitions must be verified, not just those required during normal machine operations. This is because in system operation, certain events may occur that cause the logic to assume an illegal state: power-up, brown out, line voltage glitches, etc. To test a design for proper treatment of these conditions, a method must be provided to break the feedback paths and force any desired state (i.e., illegal) into the registers. Then the machine can be sequenced and the outputs tested for correct next state generation.

All of the registers in the GAL39V18 can be preloaded, including the input, I/O, and state registers. In addition, the contents of the state and output registers can be examined in a special diagnostics mode. Programming hardware takes care of all preload timing and voltage requirements.

Input Buffers

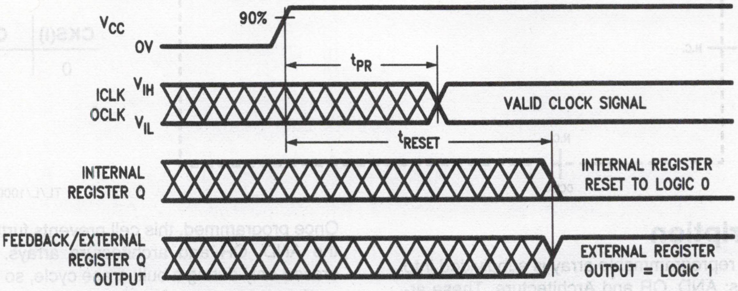
GAL devices are designed with TTL level compatible input buffers. These buffers, with their characteristically high impedance, load driving logic much less than "traditional bipolar devices." This allows for a greater fan out from the driving logic.

GAL devices do not possess active pull-ups within their input structures. As a result, National recommends that all unused inputs and TRI-STATE I/O pins be connected to another active input, V_{CC}, or GND. Doing this will tend to improve noise immunity and reduce I_{CC} for the device.

Power-Up Reset

Circuitry within the GAL39V18 provides a reset signal to all registers during power-up. All internal registers will have their Q outputs set low after a specified time (t_{RESET}). As a result, the state on the registered output pins (if they are enabled) will always be high on power-up, regardless of the programmed polarity of the output pins. This feature can greatly simplify state machine design by providing a known state on power-up.

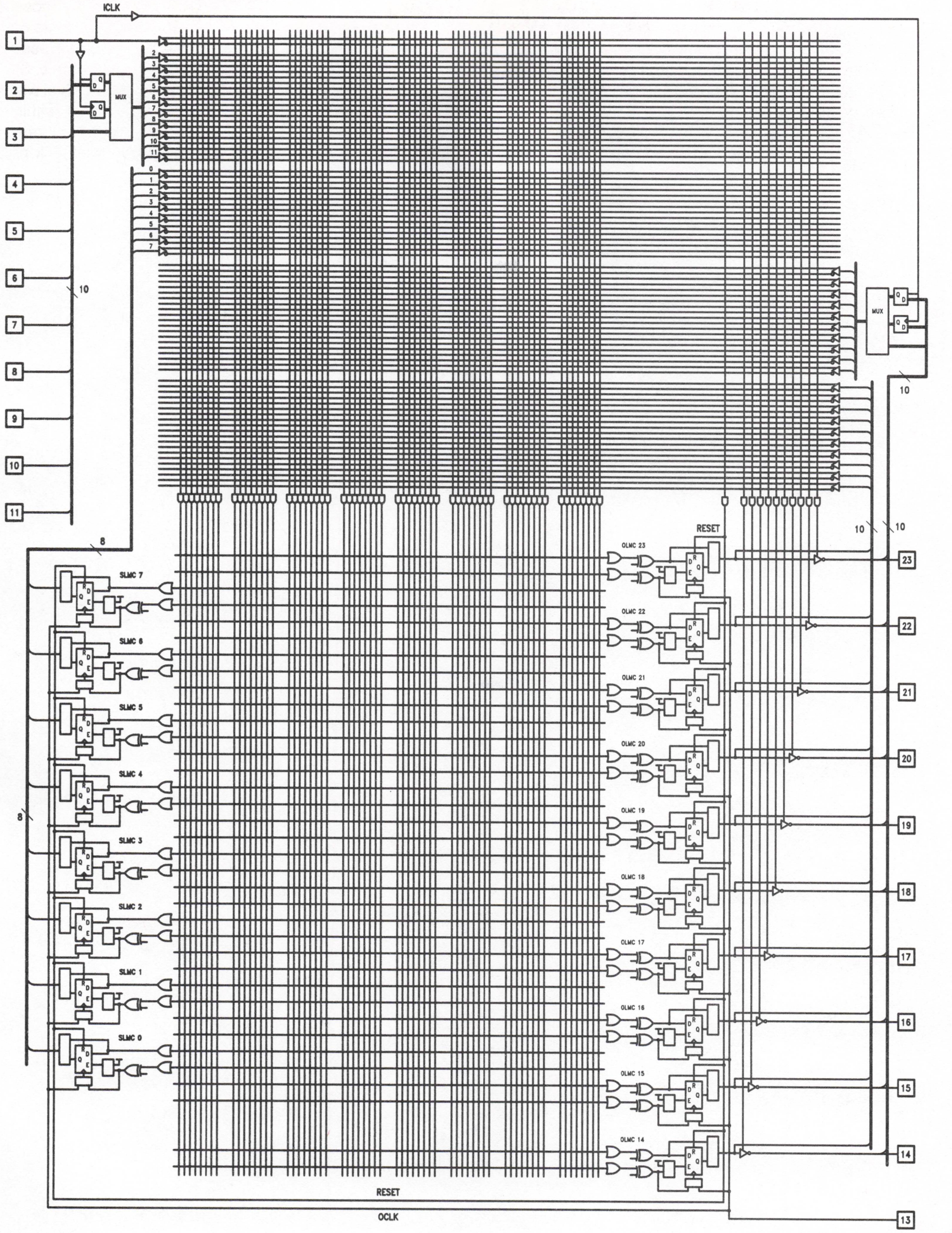
The timing diagram for power-up is shown above. Because of the asynchronous nature of system power-up, some conditions must be met to guarantee a valid power-up reset of the GAL39V18. First, the V_{CC} rise must be monotonic. Second, the clock inputs must become a proper TTL level within the specified time (t_{PR}). The registers will reset within a maximum of t_{RESET} time. As in normal system operation, avoid clocking the device until all input and feedback path setup times have been met.



V_{CC}
0V
90%
t_{PR}
ICLK
OCLK
V_{IH}
V_{IL}
VALID CLOCK SIGNAL
t_{RESET}
INTERNAL REGISTER Q
INTERNAL REGISTER RESET TO LOGIC 0
FEEDBACK/EXTERNAL REGISTER Q OUTPUT
EXTERNAL REGISTER OUTPUT = LOGIC 1

TL/L/10002-12

GAL39V18 Logic Diagram


ICLK
1
2
3
4
5
6
7
8
9
10
11
MUX
10
8
SLMC 7
SLMC 6
SLMC 5
SLMC 4
SLMC 3
SLMC 2
SLMC 1
SLMC 0
RESET
OLMC 23
OLMC 22
OLMC 21
OLMC 20
OLMC 19
OLMC 18
OLMC 17
OLMC 16
OLMC 15
OLMC 14
23
22
21
20
19
18
17
16
15
14
13
RESET
OCLK
D
Q
R
E

TL/L/10002-3

National Semiconductor

PAL10/10016P8
ECL Programmable Array Logic

General Description

The PAL1016P8/10016P8 is the first member of an ECL programmable logic device family possessing common electrical characteristics, utilizing an easily accommodated programming procedure, and produced with National Semiconductor's advanced oxide-isolated process. This family includes combinatorial, latched, and registered output devices.

These devices are fabricated using National's proven Ti-W (Titanium-Tungsten) fuse technology to allow fast, efficient, and reliable programming.

This family allows the designer to quickly implement the defined logic function by removing the fuses required to properly configure the internal gates and/or registers. Product terms with all fuses removed assume a logical high state. All devices in this series are provided with an output polarity fuse that, if removed, will permit any output to independently provide a logic low when the equation is satisfied. When these fuses are intact the outputs provide a logic true (most positive voltage level) in response to the input conditions defined by the applicable equation. All input and I/O pins have on-chip 50 kΩ pull-down resistors.

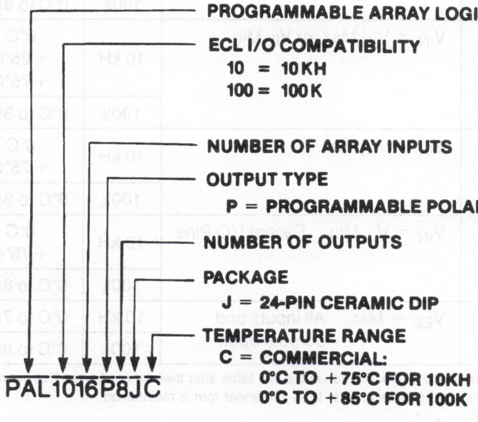
Fuse symbols have been omitted from the logic diagrams to allow the designer use of the diagrams to create fuse maps representing the programmed device.

All devices in this family can be programmed using conventional programmers. After the device has been programmed and verified, an additional fuse may be removed to inhibit further verification or programming. This "security" feature can provide a proprietary circuit which cannot easily be duplicated.

Features

- t_{PD} = 6 ns max
- Eight combinatorial outputs with programmable polarity
- Programmable replacement for conventional ECL logic
- Both 10KH and 100K I/O compatible versions
- Simplifies prototyping and board layout
- 24-pin thin DIP packages.
- Programmed on conventional TTL PLD programmers
- Security fuse to prevent direct copying
- Reliable titanium-tungsten fuses

Ordering Information


PROGRAMMABLE ARRAY LOGIC FAMILY
ECL I/O COMPATIBILITY
10 = 10KH
100 = 100K
NUMBER OF ARRAY INPUTS
OUTPUT TYPE
P = PROGRAMMABLE POLARITY
NUMBER OF OUTPUTS
PACKAGE
J = 24-PIN CERAMIC DIP
TEMPERATURE RANGE
C = COMMERCIAL:
0°C TO +75°C FOR 10KH
0°C TO +85°C FOR 100K
PAL1016P8JC

TL/L/6161-1

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias (Ambient)	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
V_{EE} Relative to V_{CC}	−7V to +0.5V
Any Input Relative to V_{CC}	V_{EE} to +0.5V
Lead Temperature (Soldering, 10 seconds)	300°C
ESD Tolerance	1000V

C_{ZAP} = 100 pF

R_{ZAP} = 1500 Ω

Test Method: Human Body Model

Test Specification: NSC SOP-5-028

Recommended Operating Conditions

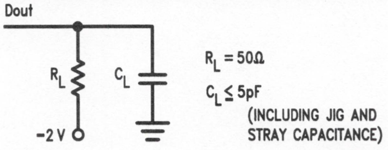
Symbol	Parameter		Min	Typ	Max	Units
V_{EE}	Supply Voltage	10 kH	−5.46	−5.2	−4.94	V
		100k	−4.73	−4.5	−4.27	
R_L	Standard 10 kH/100k Load			50		Ω
C_L	Standard 10 kH/100k Load			5		pF
T_A	Operating Ambient Temperature	10 kH	0		+75	°C
		100k	0		+85	

Electrical Characteristics

Over Recommended Operating Conditions. Output Load = 50Ω to −2.0V.

Symbol	Parameter	Conditions		T_A	Min	Max	Units
V_{IH}	High Level Input Voltage	Guaranteed input voltage high for all inputs	10 kH	0°C +25°C +75°C	−1170 −1130 −1070		mV
			100k	0°C to 85°C	−1165	−880	
V_{IL}	Low Level Input Voltage	Guaranteed input voltage low for all inputs	10 kH	0°C +25°C +75°C		−1480 −1480 −1450	mV
			100k	0°C to 85°C	−1810	−1475	
V_{OH}	High Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 kH	0°C +25°C +75°C	−1020 −980 −920	−840 −810 −735	mV
			100k	0°C to 85°C	−1025	−880	
V_{OL}	Low Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 kH	0°C +25°C +75°C	−1950 −1950 −1950	−1630 −1630 −1600	mV
			100k	0°C to 85°C	−1810	−1620	
I_{IH}	High Level Input Current	$V_{IN} = V_{IH}$ Max.	10 kH	0°C +75°C		220	µA
			100k	0°C to 85°C			
I_{IL}	Low Level Input Current	$V_{IN} = V_{IL}$ Min. Except I/O Pins	10 kH	0°C +75°C	0.5		µA
			100k	0°C to 85°C			
I_{EE}	Supply Current	V_{EE} = Max. All inputs and outputs open	10 kH	0°C to 75°C	−240		mA
			100k	0°C to 85°C			

Note: This product family has been designed to meet the specification in the test table after thermal equilibrium has been established. The circuit is in a test socket or mounted on a printed circuit board and transverse air flow greater than 500 linear fpm is maintained.



TL/L/6161-4

Switching Characteristics

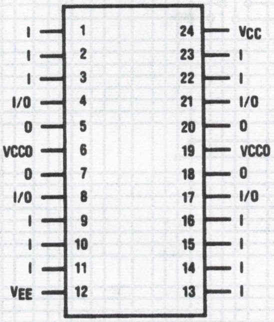
Over Recommended Operating Conditions; Output load: $R_L = 50\Omega$ to $-2.0V$, $C_L = 5$ pF to GND.

Symbol	Parameter	Test Conditions	Min	Typ	Max	Units
t_{PD}	Input to Output*			4	6	ns
t_r	Output Rise Time		0.5	1	2.5	ns
t_f	Output Fall Time		0.5	1	2.5	ns

*Measure t_{PD} at 50% points of waveforms

Connection Diagram

Dual-In-Line Package



TL/L/6161-2

Top View

PAL Design

The first step in designing a PAL device is the selection of the appropriate device to accommodate the logic equations. This is accomplished by partitioning the system into logic blocks with a defined number of inputs and outputs. Next, a device with an equal or greater I/O capability is selected to implement each logic block. The assignment of inputs and outputs to specific pins follows the device selection.

This device selection procedure is most easily accomplished with the use of computer software such as the PLAN™ package of programs by National Semiconductor Corporation, but can be done manually using the logic diagram and logic symbols provided in this document.

Specifying the Fuse Pattern

Once a device with pinout is selected, the fuse pattern may be specified. The best procedure is the use of the PLAN, or a similar software package which will create the fuse pattern from the defined logic for the device and download the pattern to a programmer. Most common device programmers are provided with an RS-232 port which accesses the data provided in JEDEC or a selected HEX format.

Logic diagrams can be translated to PAL logic diagrams if desired. Fuses left intact are indicated on the logic diagram by an "X" at the intersection of the input line and the AND gate product line. A blown fuse is not marked. The PAL logic diagrams are provided with no fuse locations marked, allowing the designer to use the diagram to manually create a fuse map. Actually, the unprogrammed device is shipped with all Xs (fuses) intact. Each fuse node is identified by a product line number and an input line number.

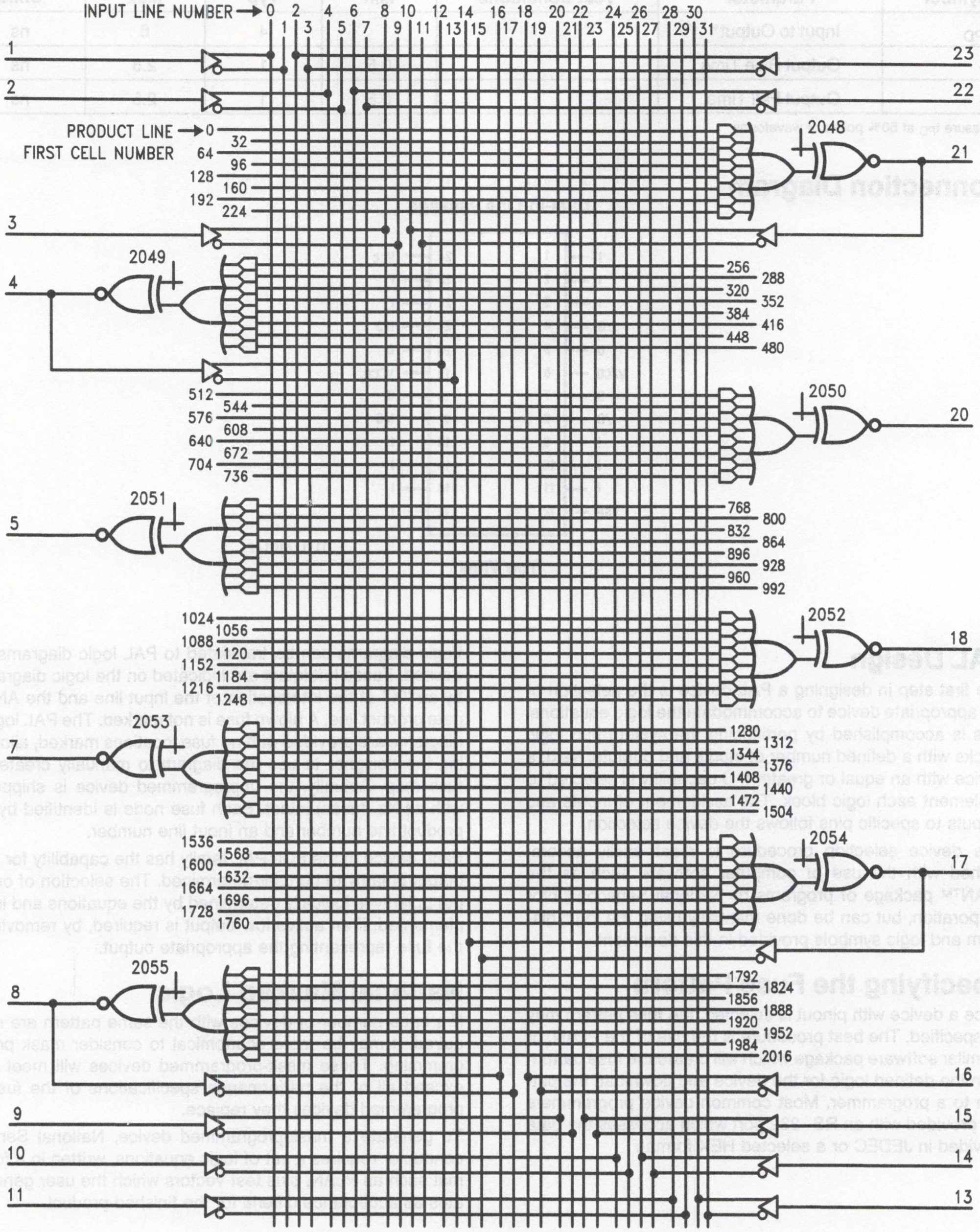
Each device in the ECL PAL family has the capability for its output polarity to be user-determined. The selection of output polarity is logically determined by the equations and implemented, if an active low output is required, by removing the fuse representing the appropriate output.

National Masked Logic

If a large number of devices with the same pattern are required, it may be more economical to consider mask programming. These mask-programmed devices will meet or exceed all of the performance specifications of the fuse-programmed devices they replace.

To generate a mask-programmed device, National Semiconductor requires a set of logic equations, written in a format such as PLAN, plus test vectors which the user generates as acceptance criteria for the finished product.

Logic Diagram PAL1016P8/PAL10016P8


INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE
FIRST CELL NUMBER
1
2
3
4
5
7
8
9
10
11
23
22
21
20
18
17
16
15
14
13
0 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
2048
2049
2050
2051
2052
2053
2054
2055

TL/L/6161-3

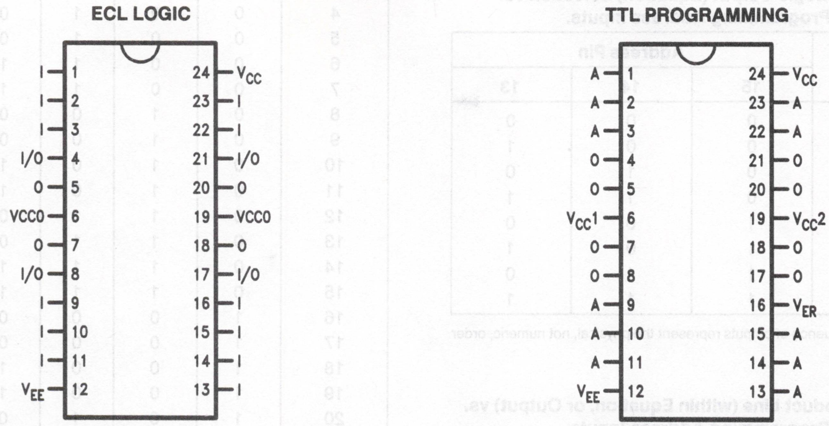
JEDEC logic array cell number = product line first cell number + input line number.

Programming Specification

This specification defines the programming and verification procedure for the first programmable logic devices in Nationals generic ECL family. The internal fuse arrays consists of 64 product lines (8 for each output), each containing 32 fuse locations (1 for each of 16 inputs and its complement) for a total of 2048 array fuses. Eight additional fuses exist to allow changing the active output polarity.

Each ECL device is programmed and verified as a 2048x1 TTL PROM. The connection diagrams in *Figure 1* illustrate the difference between the logical ECL device and the PROGRAMMABLE TTL device.

For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor.


ECL LOGIC
I 1
I 2
I 3
I/O 4
O 5
VCCO 6
O 7
I/O 8
I 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 I/O
20 O
19 VCCO
18 O
17 I/O
16 I
15 I
14 I
13 I
TTL PROGRAMMING
A 1
A 2
A 3
O 4
O 5
$V_{CC}1$ 6
O 7
O 8
A 9
A 10
A 11
V_{EE} 12
24 V_{CC}
23 A
22 A
21 O
20 O
19 $V_{CC}2$
18 O
17 O
16 V_{ER}
15 A
14 A
13 A

TL/L/6161-5

TL/L/6161-6

FIGURE 1. Connection Diagrams

Array Fuse Addressing

When programming or verifying a fuse location, the output (equation) is addressed by the 3 address pins 13, 14, and 15. The eight product lines, within the equation, are selected by the 3 address pins 9, 10, and 11. The fuse pair locations representing the logical inputs are selected by the 4 address pins 2, 3, 22, and 23, with the complementing fuse within the pair by the address pin 1. The programming address data is detailed in Tables I–III.

Table I. Logic Output (Equation) Selection vs. Programming Address Inputs.

Output Pin	Address Pin		
	15	14	13
21	0	0	0
4	0	0	1
20	0	1	0
5	0	1	1
18	1	0	0
7	1	0	1
17	1	1	0
8	1	1	1

Note that the sequence of outputs represent the physical, not numeric, order of logical outputs.

Table II. Product Line (within Equation, or Output) vs. Programming Address Inputs.

Product Pin	Address Pin		
	11	10	9
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Table III. Input Line Selection vs. Programming Address Inputs.

Input Line	Address Pin				
	23	22	3	2	1
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1
24	1	1	0	0	0
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1
28	1	1	1	0	0
29	1	1	1	0	1
30	1	1	1	1	0
31	1	1	1	1	1

Note pin 1 affects complementing fuse only.

Fuse Programming and Verification

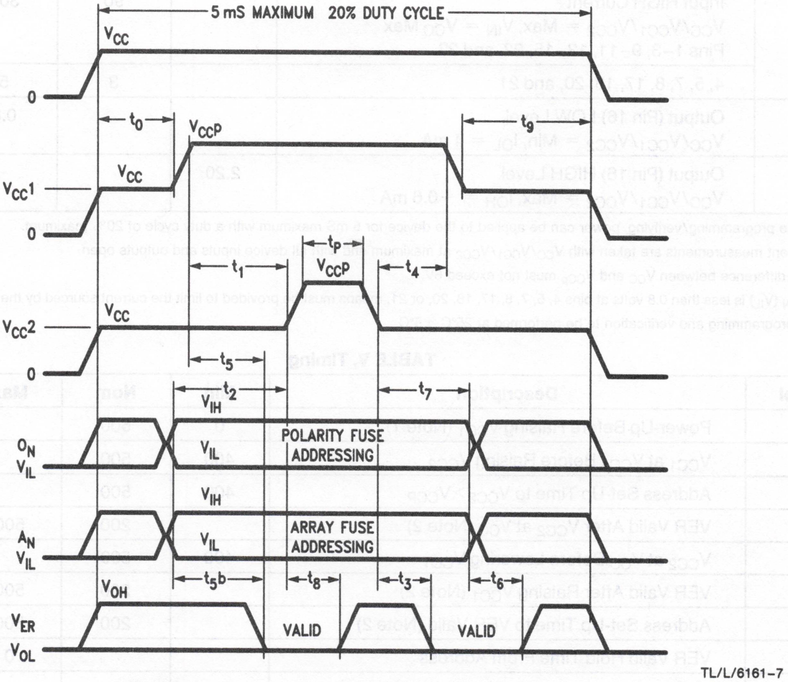
The array and output polarity fuse programming waveform diagram is shown in *Figure 2*. The 8 output pins 0_N are used only to change the polarity of the selected device output and for removing the "security" fuse. Tables 4 and 5 define the voltage and timing requirements.

Programming Procedure

1. Power is applied to the device. VCC, VCC1, and VCC2 (pins 24, 6, and 19) go to VCC. (The voltage applied to pin 24 cannot precede the voltage applied to pin 6) The output pins (4, 5, 7, 8, 17, 18, 20, and 21), are open circuited, or held at a logic low level, while programming the array.
2. After T0, VCC1 (pin 6) can be raised from 5.0 to 10.75V at a slew rate not to exceed 10V/μS, or not less than 1V/μs.
3. The 11 address inputs (pins 1–3, 9–11, 13–15, 22, and 23) will define the location of the array fuse to be opened or the applicable output pin will define the polarity fuse to be opened.
4. After VCC1 has been stable at 10.75V for period T1 and the address has been stable defining the applicable fuse location for period T2, VCC2 (pin 19) may slew from 5.0 to 10.75V at a slew rate of 1 to 10V/μS.
5. VCC2 must remain stable at 10.75V for the duration of the programming pulse (TP) before returning to 5.0V.
6. With VCC1 at 10.75V and after VCC2 has been stable at 5.0V for the period T3, VER pin (16) may be sampled. If the fuse was properly opened, a logic low level will be observed. If the fuse did not open, steps 4 through 6 may be repeated up to 15 times.
7. If additional locations are to be addressed, steps 3 through 6 must be repeated for each fuse to be opened while observing the maximum power up time and duty cycle.

Fuse Verification

Fuse verification may be performed independent of programming. As seen in *Figure 2*, with VCC1 at VCCP and VCC2 at VCC verification may occur within the defined timing constraints. (See Table V)



TL/L/6161–7

FIGURE 2. Array/Polarity Programming Diagram

TABLE IV. DC Requirements

Symbol	Description	Min	Nom	Max	Units
V_{CC}	Pin 24 Voltage While Programming or Verifying (Pin 19 Verifying) (Note 1)	4.75	5.00	5.25	V
I_{CC}	Pin 24 Current While Programming (Note 2)		200	300	mA
V_{CCp}	V_{CC1}/V_{CC2} (Pins 6 and 19) Voltage While Programming (Note 3)	10.50	10.75	11.00	V
I_{CC1}	V_{CC1} (Pin 6) Current While Programming (Note 2)		300	450	mA
I_{CC2}	VCC2 (Pin 19) Current While Programming (Note 2)		10	25	mA
VIL	Input LOW Level - If Left Open, Pins 4, 5, 7, 8, 17, 18, 20, and 21 are Held Low by Internal 50K Resistor	0		0.8	V
I_{IL}	Input LOW Current - Pins; 1–3, 9–11, 13–15, 22, and 23 $V_{CC}/V_{CC1}/V_{CC2}$ = Max, V_{IN} = 0.4V		−1.0	−1.5	mA
	4, 5, 7, 8, 17, 18, 20, and 21 (Note 4) $V_{CC}/V_{CC1}/V_{CC2}$ = Max, V_{IN} = 0.8V		−0.25	−1.5	mA
V_{IH}	Input HIGH Level	2.20		V_{CC}	V
I_{IH}	Input HIGH Current $V_{CC}/V_{CC1}/V_{CC2}$ = Max, V_{IN} = V_{CC} Max Pins 1–3, 9–11, 13–15, 22, and 23		90	300	μA
	4, 5, 7, 8, 17, 18, 20, and 21		3	5	mA
V_{OL}	Output (Pin 16) LOW Level $V_{CC}/V_{CC1}/V_{CC2}$ = Min, I_{OL} = 4 mA			0.8	V
V_{OH}	Output (Pin 16) HIGH Level $V_{CC}/V_{CC1}/V_{CC2}$ = Max, I_{OH} = −0.6 mA	2.20			V

Note 1: While programming/verifying, power can be applied to the device for 5 mS maximum with a duty cycle of 20% maximum.

Note 2: Current measurements are taken with $V_{CC}/V_{CC1}/V_{CC2}$ at maximum and with all device inputs and outputs open.

Note 3: The difference between V_{CC} and V_{CCp} must not exceed 6V.

Note 4: If V_{IN} (V_{IL}) is less than 0.8 volts at pins 4, 5, 7, 8, 17, 18, 20, or 21, means must be provided to limit the current sourced by the device pins to 10 mA.

Note 5: All programming and verification to be performed at 25°C ±5°C

TABLE V. Timing

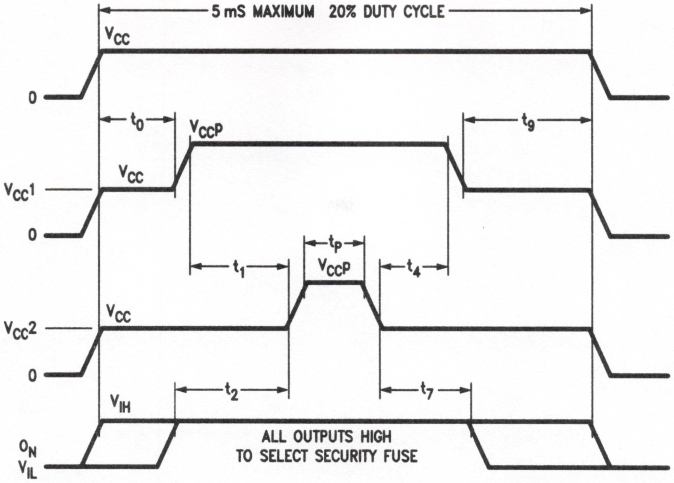
Symbol	Description	Min	Nom	Max	Units
T0	Power-Up Before Raising V_{CC1} (Note 1)	0	500		ns
T1	V_{CC1} at V_{CCp} Before Raising V_{CC2}	400	500		ns
T2	Address Set-Up Time to $V_{CC2} > V_{CCP}$	400	500		ns
T3	VER Valid After V_{CC2} at V_{CC} (Note 2)		200	500	ns
T4	V_{CC2} at V_{CC} Before Lowering V_{CC1}	400	500		ns
T5	VER Valid After Raising V_{CC1} (Note 2)		200	500	ns
T5b	Address Set-Up Time to VER Valid (Note 2)		200	500	ns
T6	VER Valid Hold Time From Address			0	ns
T7	V_{CC2} at V_{CC} Before Address Change	400	500		ns
T8	VER Valid Hold Time From $V_{CC2} > V_{CCP}$ (Note 2)	0	100		ns
T9	V_{CC1} at V_{CC} Before Power Down	0			ns
TP	Programming Pulse	10	10	30	μs

Note 1: Observe the maximum power-up time or 5 ms and duty cycle of 20% for $V_{CC}/V_{CC1}/V_{CC2}$ during programming.

Note 2: VER is valid when $V_{CC2} = V_{CC}$ and $V_{CC1} = V_{CCP}$.

Security Fuse Programming

The security fuse is opened using the same procedure as used for changing the output polarity, except all 8 outputs (pins 4, 5, 7, 8, 17, 18, 20, and 21) must be selected with the application of V_{IH}. Verification is determined by the inability to further verify the array.


5 mS MAXIMUM 20% DUTY CYCLE
V_CC
0
t_0
V_CCP
t_9
V_CC
V_CC1
0
t_P
t_1
V_CCP
t_4
V_CC
V_CC2
0
t_2
t_7
V_IH
O_N
V_IL
ALL OUTPUTS HIGH
TO SELECT SECURITY FUSE

TL/L/6161-8

FIGURE 3. Security Fuse Programming Diagram

National Semiconductor

PRELIMINARY

ECL Registered and Latched Programmable Array Logic (PAL®) Family

General Description

The registered and latched ECL PAL family consists of six device architectures, each offered in 10KH or 100K compatible versions. A maximum propagation delay of 6 ns (input to output) characterizes the performance of this ECL PAL series. The ECL PAL family utilizes National Semiconductor's advanced oxide-isolated process and proven Titanium Tungsten (Ti-W) fuse technology to provide user-programmable logic to replace conventional ECL SSI/MSI gates and flip-flops. Typical chip count reduction gained by using PAL devices is greater than 4:1.

This family allows the system engineer to customize the chip by opening fuse links to configure AND and OR gates to perform the desired logic function. Complex interconnections that previously required time-consuming layout are thus transferred from PC board to silicon where they can easily be modified during prototype checkout or production.

The PAL transfer function is the familiar sum-of-products implemented with a single array of fusible links. The PAL device incorporates a programmable AND array driving a fixed OR array. The AND term logic matrix incorporates 16 complementary inputs and 64 product terms. The 64 product terms are grouped into eight OR functions with eight product terms each. All devices in this family are provided with output polarity fuses. These fuses permit the designer to configure each output independently to produce either a logic high (by leaving the fuse intact) or a logic low (by programming the fuse) when the equation defining that output is satisfied. In addition, the ECL PAL family offers these options:

- Output registers
- Output latches
- Dual (split) clocks
- ORed (common) clocks

Product terms with all fuses programmed assume a logical high state, while product terms connected to both the true and complement of any input assume a logical low state. All product terms in an unprogrammed part are logically low. All input and I/O pins have on-chip 50 kΩ pull-down resistors. Registers consist of D-type flip-flops which are loaded in response to the low-to-high transition of the clock input(s). Latches are transparent while the enable inputs are low, and hold data while the enable inputs are high. Fuse symbols have been omitted from the logic diagrams to allow the designer use of the diagrams for logic editing.

These ECL PAL devices may be programmed on several TTL PLD programmers. Programming is accomplished with TTL voltage levels. Once the PAL is programmed and verified, an additional security fuse may be programmed to defeat verification. This feature gives the user a proprietary circuit which is difficult to copy.

Features

- High speed:
  Combinatorial and latched outputs
  t_{pd} = 6 ns max
  Registered outputs
  t_{su} = 5 ns min
  t_{clk} = 3.5 ns max
  f_{max} = 117 MHz max
- Both 10 KH and 100K I/O compatible versions
- Eight output functions with feedback; eight dedicated inputs
- Eight registered or latched outputs, or four registered or latched outputs with four combinatorial outputs
- Individually programmable polarity on all logic outputs
- Output enable gate on all registered or latched outputs
- Reliable Titanium Tungsten fuses
- Security fuse to prevent direct copying
- Programmed on conventional TTL PLD programmers
- Fully Supported by PLAN™ Software
- Packaging:
  24-pin thin DIP (0.300″)
  24-pin QUAD CERPAK

Applications

- Programmable replacement for ECL logic
- Programmable state machine
- Address or instruction decoding

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias (Ambient)	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
V_{EE} Relative to V_{CC}	−7V to +0.5V
Input Voltage	V_{EE} to +0.5V
Output Current	−50 mA
Lead Temperature (Soldering, 10 seconds)	300°C
ESD Tolerance	1000V

$C_{ZAP} = 100$ pF
$R_{ZAP} = 1500\Omega$
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Electrical Characteristics Over Recommended Operating Conditions (Note 1)

Symbol	Parameter	Conditions		T_A	Min	Max	Units
V_{IH}	High Level Input Voltage	Guaranteed Input Voltage High For All Inputs	10 KH	0°C	−1170	−840	mV
				+25°C	−1130	−810	
				+75°C	−1070	−735	
			100K	0°C to +85°C	−1165	−880	
V_{IL}	Low Level Input Voltage	Guaranteed Input Voltage Low For All Inputs	10 KH	0°C	−1950	−1480	mV
				+25°C	−1950	−1480	
				+75°C	−1950	−1450	
			100K	0°C to +85°C	−1810	−1475	
V_{OH}	High Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 KH	0°C	−1020	−840	mV
				+25°C	−980	−810	
				+75°C	−920	−735	
			100K	0°C to +85°C	−1025	−880	
V_{OL}	Low Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 KH	0°C	−1950	−1630	mV
				+25°C	−1950	−1630	
				+75°C	−1950	−1600	
			100K	0°C to +85°C	−1810	−1620	
I_{IH}	High Level Input Current (Note 4)	$V_{IN} = V_{IH}$ Max.	10 KH	0°C +75°C		220	μA
			100K	0°C to +85°C			
I_{IL}	Low Level Input Current	$V_{IN} = V_{IL}$ Min. Except I/O Pins	10 KH	0°C +75°C	0.5		μA
			100K	0°C to +85°C			
I_{EE}	Supply Current	V_{EE} = Min. All Inputs and Outputs Open	LD8, LD4, RD4, RC4		−260		mA
			RD8, RC8		−280		

Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Units
V_{EE}	Supply Voltage	10 KH	−5.46	−5.2	−4.94	V
		100K	−4.73	−4.5	−4.27	
T	Operating Temperature (Note 2)	10 KH	0		+75	°C
		100K	0		+85	
R_L	Standard 10 KH/100K Load			50		Ω
C_L	Standard 10 KH/100K Load			5		pF
t_{SU}	Setup Time of Input or Feedback (Note 3)		5			ns
t_H	Input Hold Time (Note 3)		0			ns
t_W	Clock or Enable Pulse Width (Note 3)		4			ns
t_{WMR}	Master Reset Pulse Width (Note 3)		4			ns

Note 1: This product family has been designed to meet the specification in the test table after thermal equilibrium has been established. The circuit is in a test socket or mounted on a printed circuit board and transverse air flow greater than 500 linear fpm is maintained.

Note 2: Operating temperatures for circuits in Dual-In-Line packages are specified as ambient temperatures (T_A) with circuits mounted in socket or printed circuit board and transverse airflow exceeding 500 linear feet per minute. Operating temperatures for circuits packaged in QUAD CERPAK are specified as case temperatures (T_C).

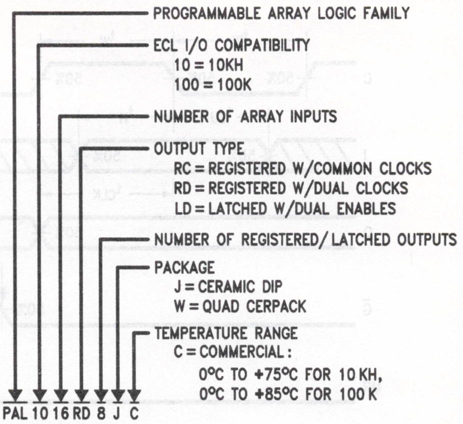
Note 3: Applies to registered and latched outputs.

Note 4: Except for clock inputs (350 μA) and MR (1 mA).

Ordering Information

The 12 products in the family are differentiated by their logic level compatibility (10 KH or 100K), by their output type (registered, latched or mixed combinatorial) and by their clock configuration (Common ORed or Dual clocks). The family consists of the following products:

Part Number	Description
PAL1016RC8 PAL10016RC8	8 Registered Outputs with Common ORed Clock
PAL1016RD8 PAL10016RD8	8 Registers with Dual Clocks (4 Registers Each)
PAL1016RC4 PAL10016RC4	4 Registers (Common Clock) Plus 4 Combinatorial I/Os
PAL1016RD4 PAL10016RD4	4 Registers (Dual Clocks) Plus 4 Combinatorial I/Os
PAL1016LD8 PAL10016LD8	8 Latches with Dual Clocks (4 Latches Each)
PAL1016LD4 PAL10016LD4	4 Latches (Dual Clocks) Plus 4 Combinatorial I/Os


PROGRAMMABLE ARRAY LOGIC FAMILY
ECL I/O COMPATIBILITY
10 = 10KH
100 = 100K
NUMBER OF ARRAY INPUTS
OUTPUT TYPE
RC = REGISTERED W/COMMON CLOCKS
RD = REGISTERED W/DUAL CLOCKS
LD = LATCHED W/DUAL ENABLES
NUMBER OF REGISTERED/ LATCHED OUTPUTS
PACKAGE
J = CERAMIC DIP
W = QUAD CERPACK
TEMPERATURE RANGE
C = COMMERCIAL:
0°C TO +75°C FOR 10 KH,
0°C TO +85°C FOR 100 K
PAL 10 16 RD 8 J C

TL/L/8765-1

Switching Characteristics Over Recommended Operating Conditions

Output Load: $R_L = 50\Omega$ to $-2.0V$, $C_L = 5$ pF to GND

Symbol	Parameter	Measured		Min	Max	Units
		From	To			
t_{CLK} (Note 1)	Clock to Output or Feedback	C_n ↑	Q		3.5	ns
t_{LE} (Note 2)	Enable to Output or Feedback	$\overline{L_n}$ ↓	Q		3.5	ns
t_{PD} (Note 2, 3)	Input or Feedback to Output	I	Q or I/O		6	ns
t_{PLH} (Note 1, 2)	Output Enable	$\overline{G}$ ↓	Q ↑		4.0	ns
t_{PHL} (Note 1, 2)	Output Disable	$\overline{G}$ ↑	Q ↓		4.0	ns
t_{MR} (Note 1, 2)	Master Reset to Output	MR ↑	Q ↓		5.5	ns
f_{MAX} (Note 1, 4)	Maximum Frequency				117	MHz
t_r	Output Rise Time	Measured Between 20% and 80% points		0.5	2.5	ns
t_f	Output Fall Time			0.5	2.5	ns

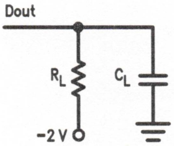
Note 1: Applies to registered outputs.

Note 2: Applies to latched outputs.

Note 3: Applies to combinatorial outputs.

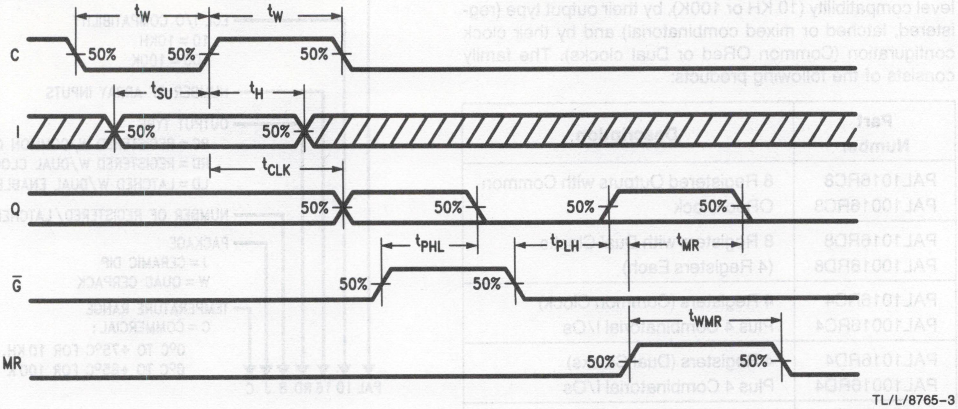
Note 4: $f_{MAX} = (t_{SU} + t_{CLK})^{-1}$

Test Load


Dout
R_L
C_L
−2 V

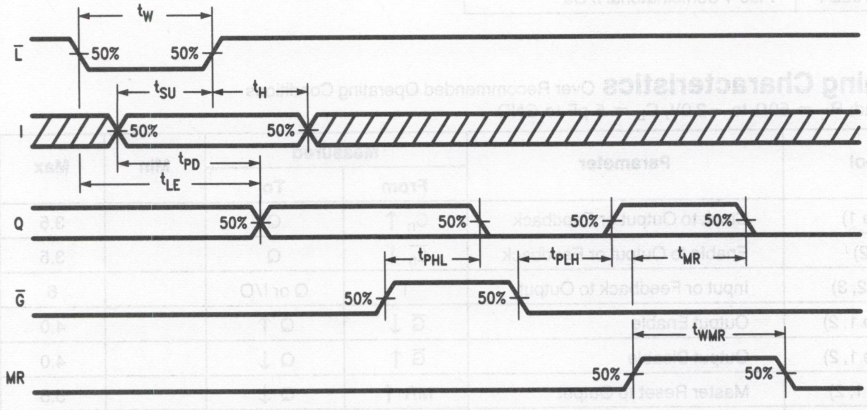
TL/L/8765-2

Timing Waveform—All Registered Outputs


C
I
Q
G̅
MR
tW
tW
tSU
tH
tCLK
tPHL
tPLH
tMR
tWMR
50%

TL/L/8765-3

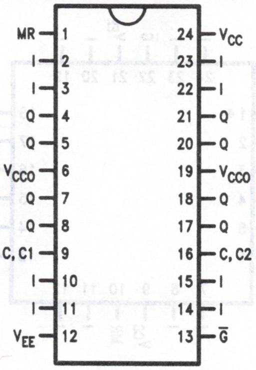
Timing Waveform—All Latched Outputs


L̅
I
Q
G̅
MR
tW
tSU
tH
tPD
tLE
tPHL
tPLH
tMR
tWMR
50%

TL/L/8765-4

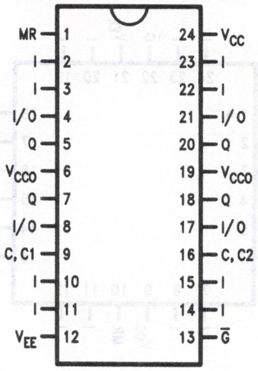
Connection Diagrams (24-pin Dual-In-Line Packages)

PAL1016RC8/PAL10016RC8
PAL1016RD8/PAL10016RD8


MR 1
I 2
I 3
Q 4
Q 5
V_{CCO} 6
Q 7
Q 8
C, C1 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 Q
20 Q
19 V_{CCO}
18 Q
17 Q
16 C, C2
15 I
14 I
13 $\overline{G}$

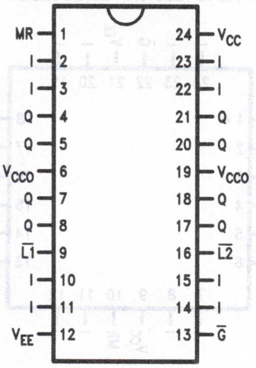
TL/L/8765-5

PAL1016RC4/PAL10016RC4
PAL1016RD4/PAL10016RD4


MR 1
I 2
I 3
I/O 4
Q 5
V_{CCO} 6
Q 7
I/O 8
C, C1 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 I/O
20 Q
19 V_{CCO}
18 Q
17 I/O
16 C, C2
15 I
14 I
13 $\overline{G}$

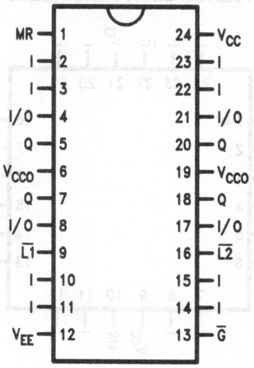
TL/L/8765-6

PAL1016LD8/PAL10016LD8


MR 1
I 2
I 3
Q 4
Q 5
V_{CCO} 6
Q 7
Q 8
$\overline{L1}$ 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 Q
20 Q
19 V_{CCO}
18 Q
17 Q
16 $\overline{L2}$
15 I
14 I
13 $\overline{G}$

TL/L/8765-7

PAL1016LD4/PAL10016LD4


MR 1
I 2
I 3
I/O 4
Q 5
V_{CCO} 6
Q 7
I/O 8
$\overline{L1}$ 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 I/O
20 Q
19 V_{CCO}
18 Q
17 I/O
16 $\overline{L2}$
15 I
14 I
13 $\overline{G}$

TL/L/8765-8

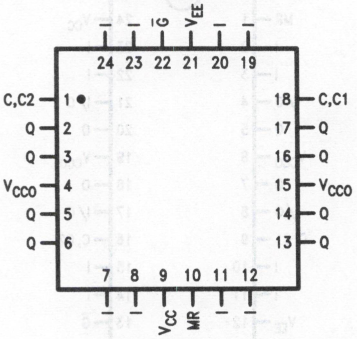
Pin Descriptions

Pin	Description
I	Eight dedicated inputs to logic array.
Q	Four or eight outputs from registered or latched logic functions.
I/O	Four outputs from combinatorial logic functions on 'RC4, 'RD4 and 'LD4. Pin signal fed back as input into logic array. Pin can be used as input or output.
C, C1	Clock input for registers on output pins* 4, 5, 7 and 8 on dual-clock devices; ORed with C2 to control all registers on common-clock devices. Data is written into registers on rising edge of clock.
C, C2	Clock input for registers on output pins* 17, 18, 20 and 21 on dual-clock devices; ORed with C1 to control all registers on common-clock devices. Data is written into registers on rising edge of clock.
$\overline{L1}$	Latch enable input for latches on output pins* 4, 5, 7 and 8. Latches are transparent (data written into latch) while the enable signal is low.
$\overline{L2}$	Latch enable input for latches on output pins* 17, 18, 20 and 21. Latches are transparent (data written into latch) while the enable signal is low.
MR	Master Reset input. Asynchronously resets all registers or latches to the low state while MR is high (overrides clock and latch enable).
$\overline{G}$	Output enable input. Enables output drivers while $\overline{G}$ is low; forces all registered or latched output drivers to the low state as long as $\overline{G}$ is high. Register or latch contents and feedbacks are not affected. Combinatorial outputs are not affected.
V_{EE}	Supply voltage.
V_{CC}	Ground for internal circuitry.
V_{CCO}	Ground for output drivers (4 outputs per V_{CCO}).

*Corresponds to DIP pinout

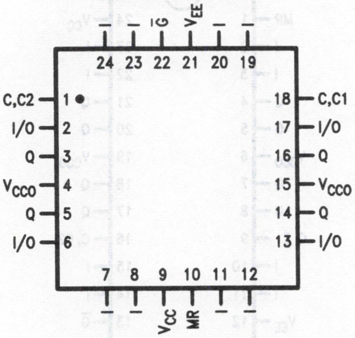
Connection Diagrams (24-pin Quad Cerpak)

PAL1016RC8/PAL10016RC8
PAL1016RD8/PAL10016RD8


I
I
$\overline{G}$
V_{EE}
I
I
24 23 22 21 20 19
C,C2 1
Q 2
Q 3
V_{CCO} 4
Q 5
Q 6
18 C,C1
17 Q
16 Q
15 V_{CCO}
14 Q
13 Q
7 8 9 10 11 12
I
I
V_{CC}
MR
I
I

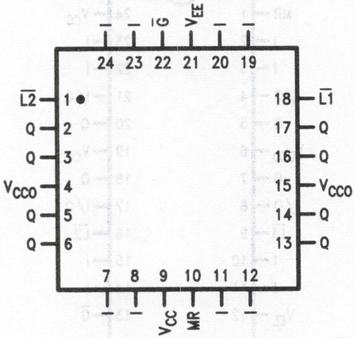
TL/L/8765-21

PAL1016RC4/PAL10016RC4
PAL1016RD4/PAL10016RD4


I
I
$\overline{G}$
V_{EE}
I
I
24 23 22 21 20 19
C,C2 1
I/O 2
Q 3
V_{CCO} 4
Q 5
I/O 6
18 C,C1
17 I/O
16 Q
15 V_{CCO}
14 Q
13 I/O
7 8 9 10 11 12
I
I
V_{CC}
MR
I
I

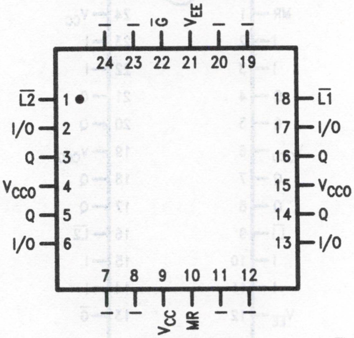
TL/L/8765-22

PAL1016LD8/PAL10016LD8


I
I
$\overline{G}$
V_{EE}
I
I
24 23 22 21 20 19
$\overline{L2}$ 1
Q 2
Q 3
V_{CCO} 4
Q 5
Q 6
18 $\overline{L1}$
17 Q
16 Q
15 V_{CCO}
14 Q
13 Q
7 8 9 10 11 12
I
I
V_{CC}
MR
I
I

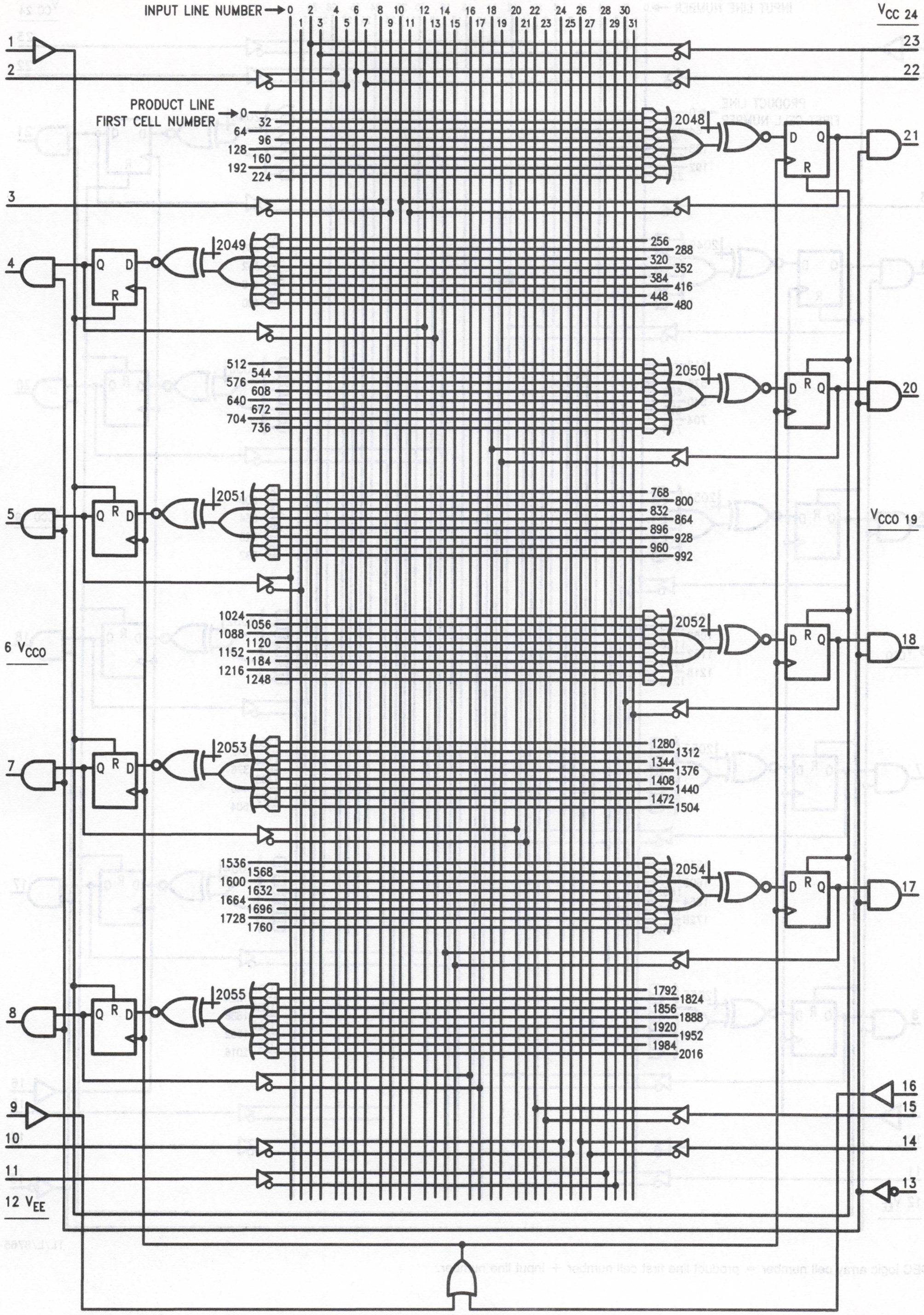
TL/L/8765-23

PAL1016LD4/PAL10016LD4


I
I
$\overline{G}$
V_{EE}
I
I
24 23 22 21 20 19
$\overline{L2}$ 1
I/O 2
Q 3
V_{CCO} 4
Q 5
I/O 6
18 $\overline{L1}$
17 I/O
16 Q
15 V_{CCO}
14 Q
13 I/O
7 8 9 10 11 12
I
I
V_{CC}
MR
I
I

TL/L/8765-24

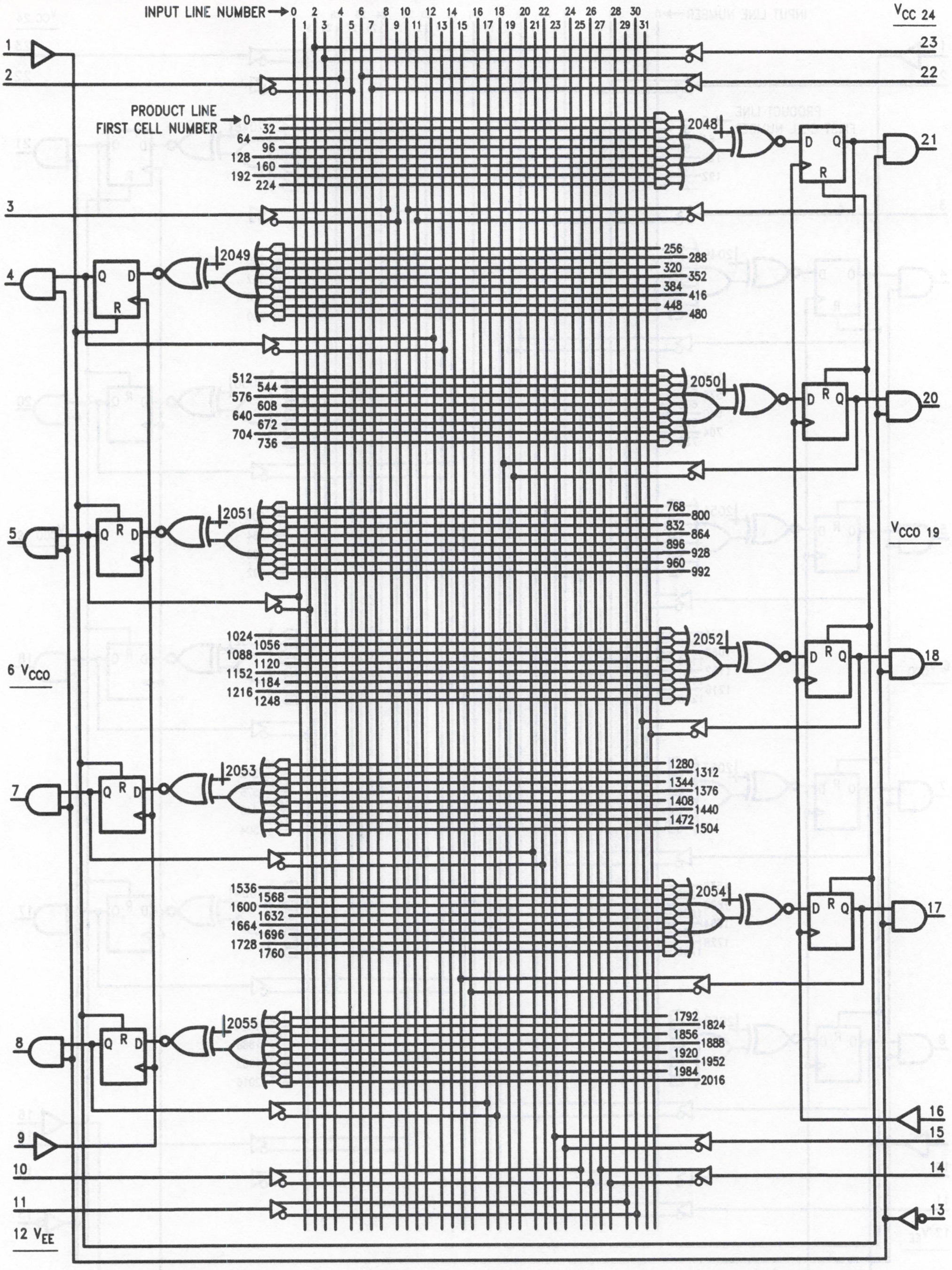
Logic Diagram PAL1016RC8/PAL10016RC8


INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE FIRST CELL NUMBER
0 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
2048 2049 2050 2051 2052 2053 2054 2055
D Q R
1 2 3 4 5 6 V_{CCO} 7 8 9 10 11 12 V_{EE}
V_{CC} 24 23 22 21 20 V_{CCO} 19 18 17 16 15 14 13

TL/L/8765-9

JEDEC logic array cell number = product line first cell number + input line number.

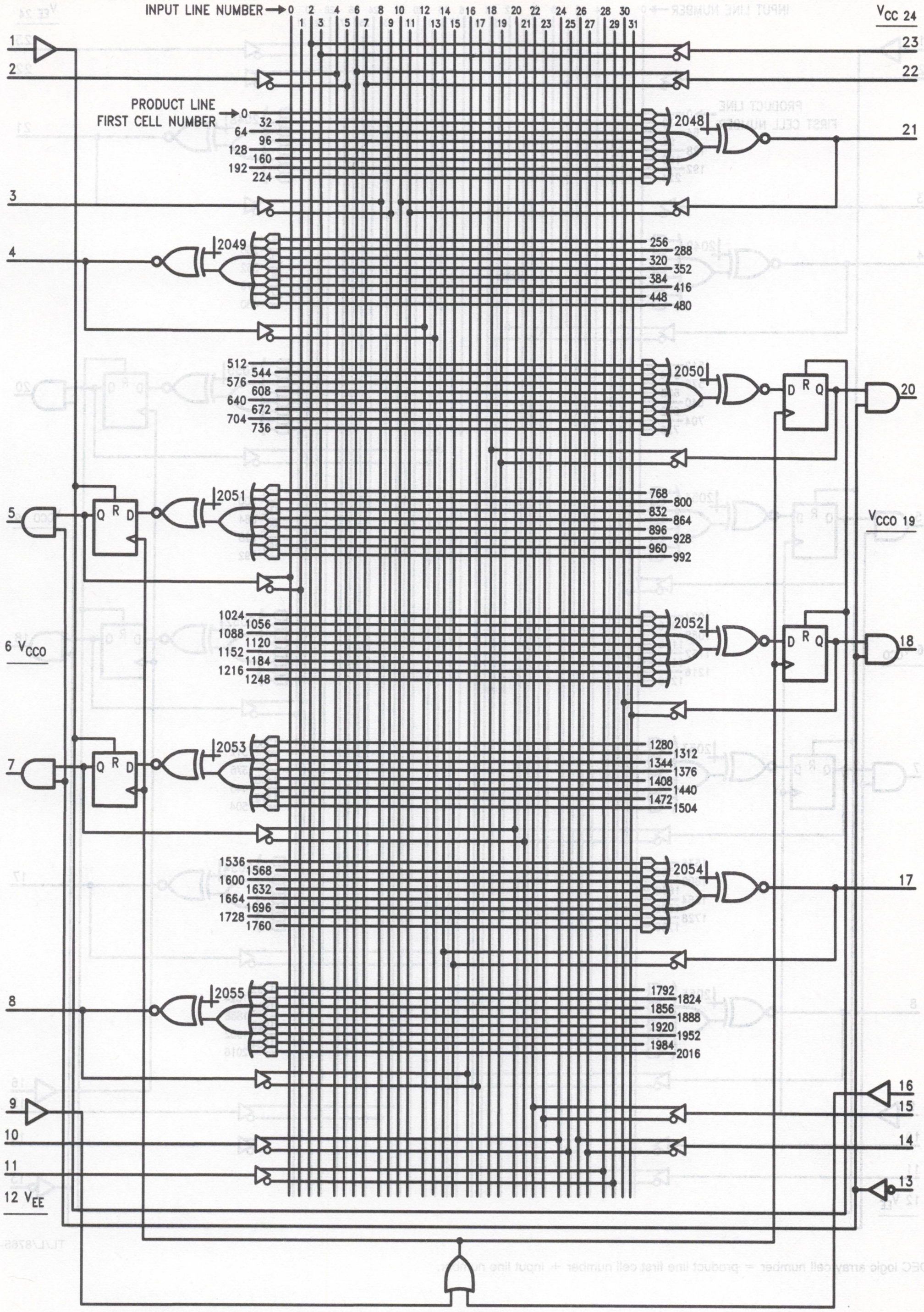
Logic Diagram PAL1016RD8/PAL10016RD8


INPUT LINE NUMBER → 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE FIRST CELL NUMBER → 0
32
64
96
128
160
192
224
2048
256
288
320
352
384
416
448
480
2049
512
544
576
608
640
672
704
736
2050
768
800
832
864
896
928
960
992
2051
1024
1056
1088
1120
1152
1184
1216
1248
2052
1280
1312
1344
1376
1408
1440
1472
1504
2053
1536
1568
1600
1632
1664
1696
1728
1760
2054
1792
1824
1856
1888
1920
1952
1984
2016
2055
D
Q
R
1
2
3
4
5
6 V_{CCO}
7
8
9
10
11
12 V_{EE}
V_{CC} 24
23
22
21
20
V_{CCO} 19
18
17
16
15
14
13

TL/L/8765-10

JEDEC logic array cell number = product line first cell number + input line number.

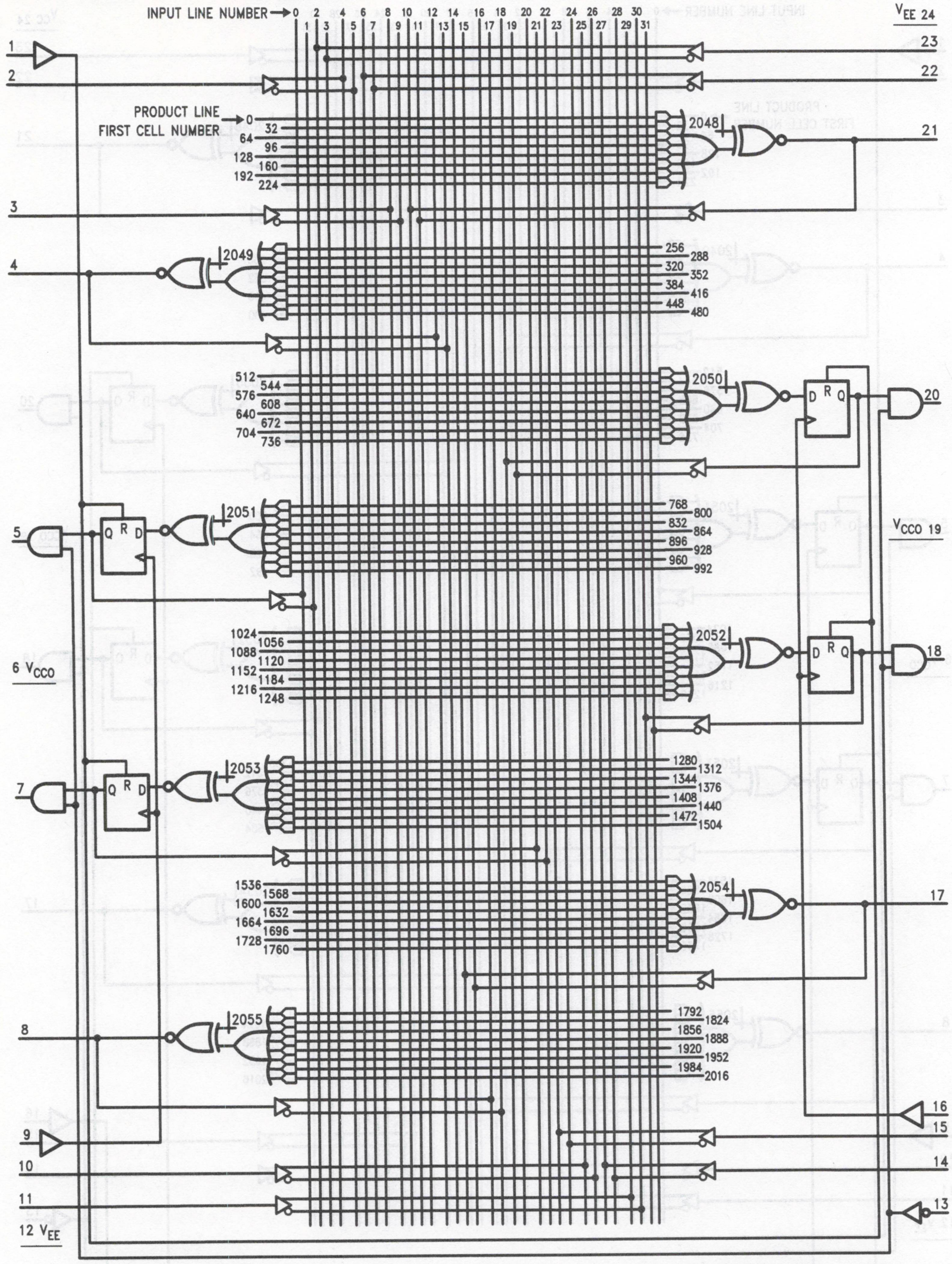
Logic Diagram PAL1016RC4/PAL10016RC4


INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE
FIRST CELL NUMBER
0 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
2048 2049 2050 2051 2052 2053 2054 2055
D R Q
1
2
3
4
5
6 V_{CCO}
7
8
9
10
11
12 V_{EE}
V_{CC} 24
23
22
21
20
V_{CCO} 19
18
17
16
15
14
13

TL/L/8765–11

JEDEC logic array cell number = product line first cell number + input line number.

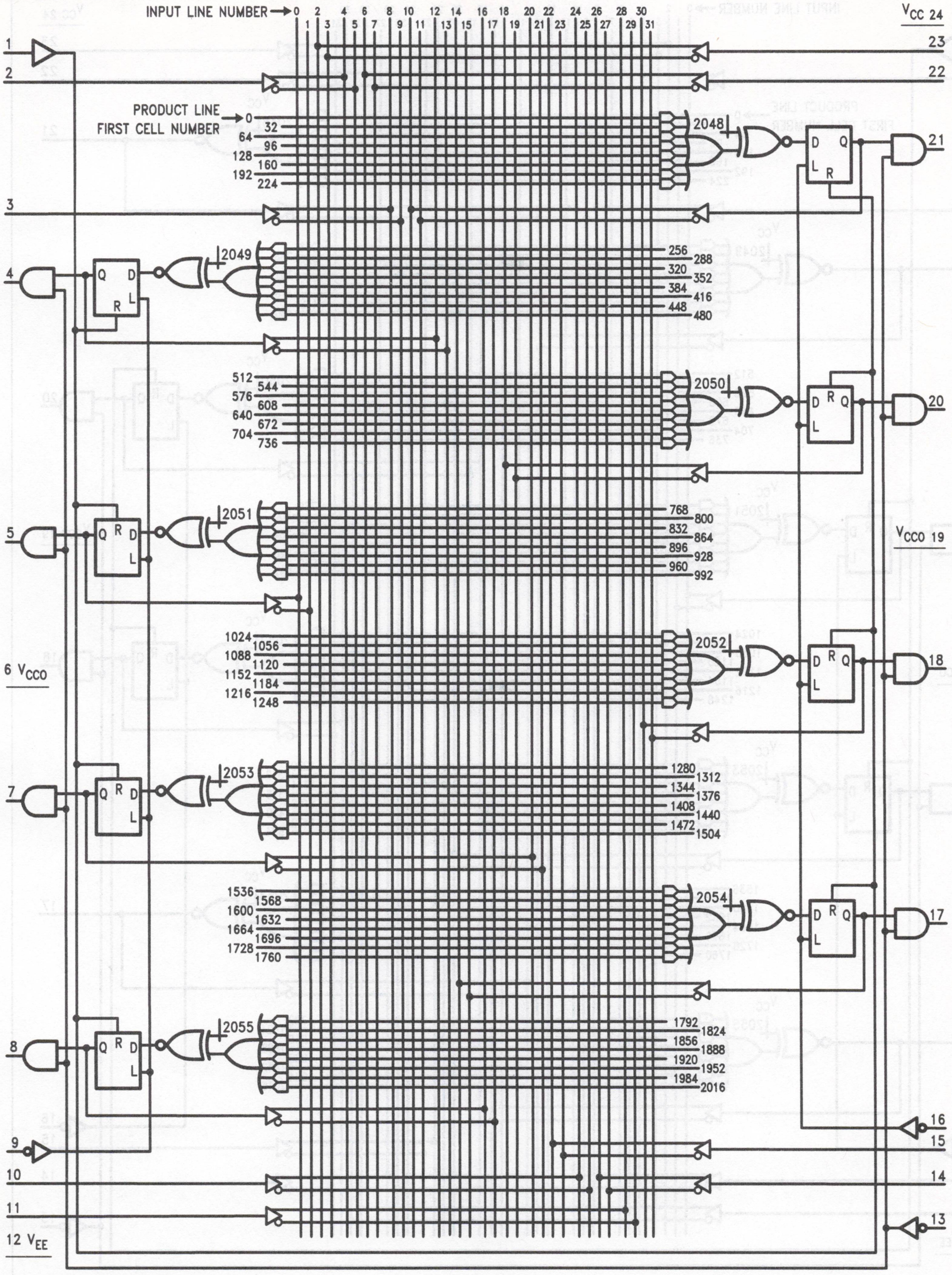
Logic Diagram PAL1016RD4/PAL10016RD4


INPUT LINE NUMBER
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
PRODUCT LINE
FIRST CELL NUMBER
0
32
64
96
128
160
192
224
256
288
320
352
384
416
448
480
512
544
576
608
640
672
704
736
768
800
832
864
896
928
960
992
1024
1056
1088
1120
1152
1184
1216
1248
1280
1312
1344
1376
1408
1440
1472
1504
1536
1568
1600
1632
1664
1696
1728
1760
1792
1824
1856
1888
1920
1952
1984
2016
2048
2049
2050
2051
2052
2053
2054
2055
1
2
3
4
5
6 V_{CCO}
7
8
9
10
11
12 V_{EE}
13
14
15
16
17
18
V_{CCO} 19
20
21
22
23
V_{EE} 24
D
R
Q

TL/L/8765-12

JEDEC logic array cell number = product line first cell number + input line number.

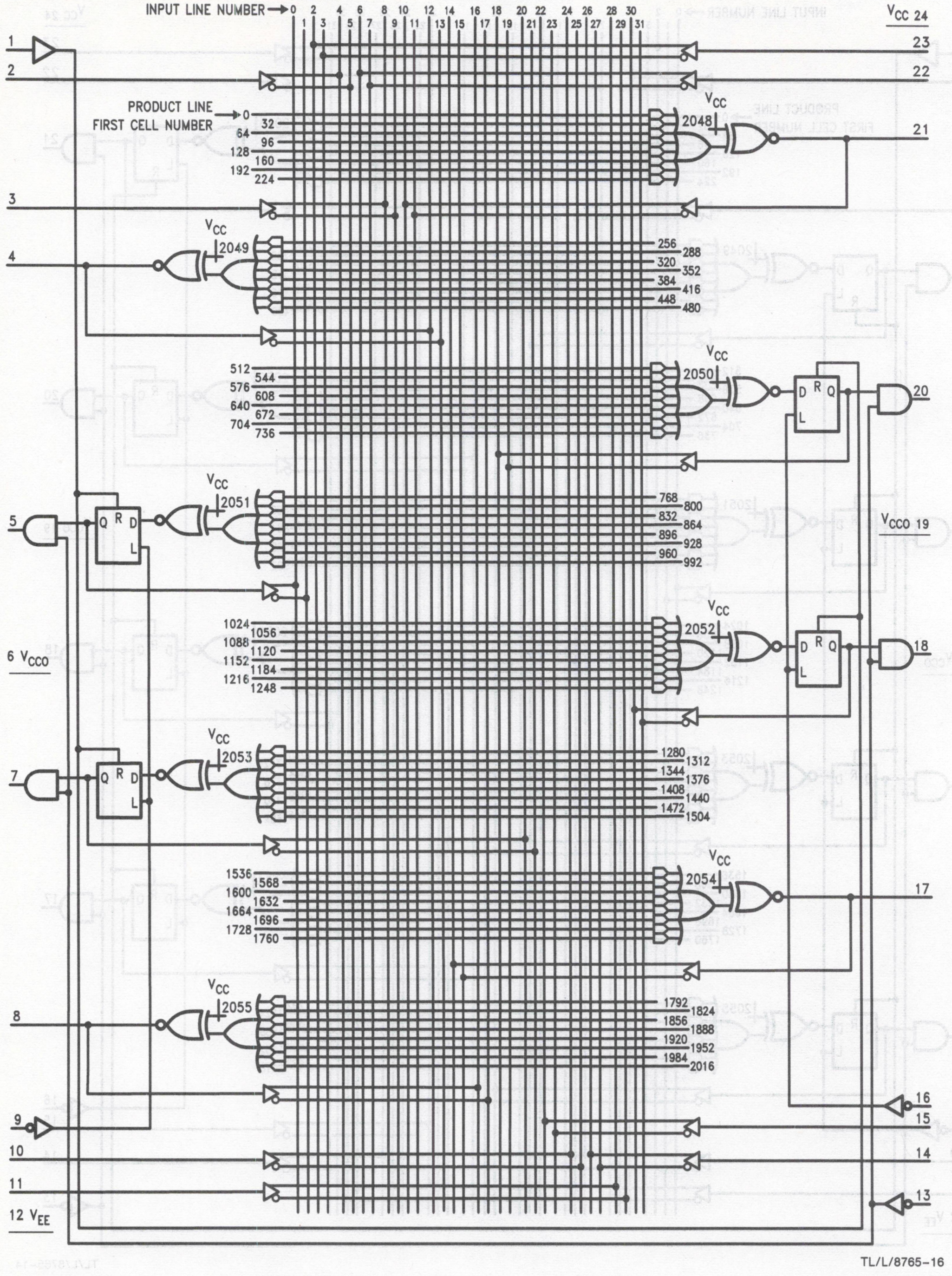
Logic Diagram PAL1016LD8/PAL10016LD8


INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE FIRST CELL NUMBER
0 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
2048 2049 2050 2051 2052 2053 2054 2055
D Q L R
1
2
3
4
5
6 V_{CCO}
7
8
9
10
11
12 V_{EE}
13
14
15
16
17
18
V_{CCO} 19
20
21
22
23
V_{CC} 24

TL/L/8765–14

JEDEC logic array cell number = product line first cell number + input line number.

Logic Diagram PAL1016LD4/PAL10016LD4


INPUT LINE NUMBER
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
PRODUCT LINE
FIRST CELL NUMBER
VCC 24
1
2
3
4
5
6 VCCO
7
8
9
10
11
12 VEE
23
22
21
20
VCCO 19
18
17
16
15
14
13
VCC
2048
2049
2050
2051
2052
2053
2054
2055
D
R
Q
L
0
32
64
96
128
160
192
224
256
288
320
352
384
416
448
480
512
544
576
608
640
672
704
736
768
800
832
864
896
928
960
992
1024
1056
1088
1120
1152
1184
1216
1248
1280
1312
1344
1376
1408
1440
1472
1504
1536
1568
1600
1632
1664
1696
1728
1760
1792
1824
1856
1888
1920
1952
1984
2016

TL/L/8765-16

JEDEC logic array cell number = product line first cell number + input cell number.

Functional Testing

As with all field-programmable devices, the user of ECL PAL devices provides the final manufacturing step. While National's PAL devices undergo extensive testing when they are manufactured, their logic function can be fully tested only after they have been programmed to the user's pattern.

To ensure that the programmed PAL devices will operate properly in your system, National Semiconductor (along with most other manufacturers of PAL devices) strongly recommends that PAL devices be functionally tested before they are installed in your system. Even though the number of post-programming functional failures is small, testing the logic function of the PAL devices before they reach system assembly will save board debugging and rework costs. Refer to National Semiconductor's Application Note #351 and the *Programmable Logic Design Guide* for more information about the functional testing of PAL devices.

For a list of current programming support tools for ECL PAL devices, please contact your local National Semiconductor sales office.

National Semiconductor

PAL10/10016P4A
4 ns ECL Programmable Array Logic

General Description

The PAL1016P4A and PAL10016P4A are members of the National Semiconductor ECL PAL® family. The PAL10/10016P4A is a functional subset of the PAL10/10016P8 (6 ns tpd) and is compatible in pinout, JEDEC map format, and programming algorithm. The ECL PAL family utilizes National Semiconductor's advanced oxide-isolated process and proven Titanium-Tungsten (Ti-W) fuse technology to provide user-programmable logic to replace conventional ECL SSI/MSI gates and flip-flops. Typical chip count reduction gained by using PAL devices is greater than 4:1.

This family allows the systems engineer to customize his chip by opening fuse links to configure AND and OR gates to perform his desired logic function. Complex interconnections that previously required time-consuming layout are thus transferred from PC board to silicon where they can easily be modified during prototype checkout or production.

The PAL transfer function is the familiar sum-of-products implemented with a single array of fusible links. The PAL device incorporates a programmable AND array driving a fixed OR array. The AND term logic matrix incorporates 16 complementary inputs and 32 product terms. The 32 product terms are grouped into four OR functions with eight product terms each. All devices in this series are provided with output polarity fuses. These fuses permit the designer to configure each output independently to provide either a logic true (by leaving the fuse intact) or a logic false (by programming the fuse) when the equation defining that output is satisfied.

Product terms with all fuses programmed assume a logical high state, while product terms connected to both the true and complement of any input assume a logical low state. All product terms in an unprogrammed part are logically low.

Fuse symbols have been omitted from the logic diagrams to allow the designer use of the diagrams for logic editing.

These ECL PAL devices may be programmed on many PLD programmers. Programming is accomplished using TTL voltage levels. Once programmed and verified, an additional fuse may be programmed to disable further verification. This feature gives the user a proprietary circuit which is difficult to copy.

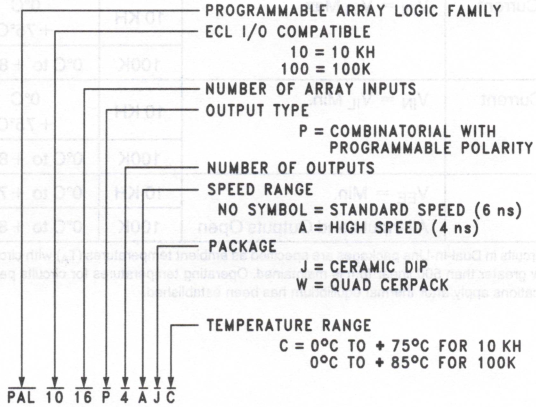
Features

- High speed:
  Combinatorial outputs
  tpd = 4 ns max
- Both 10 KH and 100K I/O compatible versions
- Four output functions; sixteen dedicated inputs
- Individually programmable polarity for all logic outputs
- Reliable titanium-tungsten fuses
- Security fuse to prevent direct copying
- Programmed on many PLD programmers
- Fully Supported by PLAN™ Software
- Packaging:
  24-pin thin DIP (0.300″)
  24-pin QUAD CERPAK

Applications

- Programmable replacement for ECL logic
- Address or instruction decoding

Ordering Information


PROGRAMMABLE ARRAY LOGIC FAMILY
ECL I/O COMPATIBLE
10 = 10 KH
100 = 100K
NUMBER OF ARRAY INPUTS
OUTPUT TYPE
P = COMBINATORIAL WITH PROGRAMMABLE POLARITY
NUMBER OF OUTPUTS
SPEED RANGE
NO SYMBOL = STANDARD SPEED (6 ns)
A = HIGH SPEED (4 ns)
PACKAGE
J = CERAMIC DIP
W = QUAD CERPACK
TEMPERATURE RANGE
C = 0°C TO + 75°C FOR 10 KH
0°C TO + 85°C FOR 100K
PAL 10 16 P 4 A J C

TL/L/9138–1

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
V_{EE} Relative to V_{CC}	−7V to +0.5V
Any Input Relative to V_{CC}	V_{EE} to +0.5V
Lead Temperature (Soldering, 10 seconds)	300°C
ESD Tolerance	1000V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Units
V_{EE}	Supply Voltage	10 KH	−5.46	−5.2	−4.94	V
		100K	−4.73	−4.5	−4.27	
T	Operating Temperature (Note)	10 KH	0		+75	°C
		100K	0		+85	

DC Electrical Characteristics Over Recommended Operating Conditions

Output Load = 50Ω to −2.0V

Symbol	Parameter	Conditions		T_A	Min	Max	Units
V_{IH}	High Level Input Voltage	Guaranteed Input Voltage High For All Inputs	10 KH	0°C	−1170	−840	mV
				+25°C	−1130	−810	
				+75°C	−1070	−735	
			100K	0°C to +85°C	−1165	−880	
V_{IL}	Low Level Input Voltage	Guaranteed Input Voltage Low For All Inputs	10 KH	0°C	−1950	−1480	mV
				+25°C	−1950	−1480	
				+75°C	−1950	−1450	
			100K	0°C to +85°C	−1810	−1475	
V_{OH}	High Level Output Voltage	V_{IN} = V_{IH} Max. or V_{IL} Min.	10 KH	0°C	−1020	−840	mV
				+25°C	−980	−810	
				+75°C	−920	−735	
			100K	0°C to +85°C	−1025	−880	
V_{OL}	Low Level Output Voltage	V_{IN} = V_{IH} Max. or V_{IL} Min.	10 KH	0°C	−1950	−1630	mV
				+25°C	−1950	−1630	
				+75°C	−1950	−1600	
			100K	0°C to +85°C	−1810	−1620	
I_{IH}	High Level Input Current	V_{IN} = V_{IH} Max.	10 KH	0°C +75°C		220	µA
			100K	0°C to +85°C			
I_{IL}	Low Level Input Current	V_{IN} = V_{IL} Min.	10 KH	0°C +75°C	0.5		µA
			100K	0°C to +85°C			
I_{EE}	Supply Current	V_{EE} = Min. All Inputs and Outputs Open	10 KH	0°C to +75°C	−220		mA
			100K	0°C to +85°C			

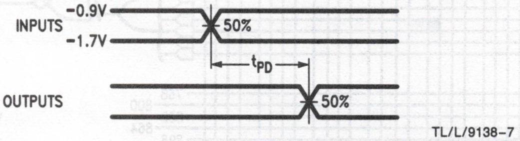
Note: Operating temperatures for circuits in Dual-In-Line packages are specified as ambient temperatures (T_A) with circuits in a test socket or mounted on a printed circuit board and transverse air flow greater than 500 linear fpm is maintained. Operating temperatures for circuits packaged in QUAD CERPAK are specified as case temperatures (T_C). All specifications apply after thermal equilibrium has been established.

Switching Characteristics

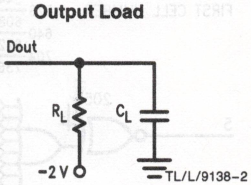
Over Recommended Operating Conditions, Output load: $R_L = 50\Omega$ to $-2.0V$, $C_L = 5$ pF to GND

Symbol	Parameter	Measured Test Conditions	Min	Max	Units
t_{PD}	Input to Output	Measured at 50% points		4	ns
t_r	Output Rise Time	Measured between 20% and 80% points	0.5	2.5	ns
t_f	Output Fall Time		0.5	2.5	ns

Timing Measurements

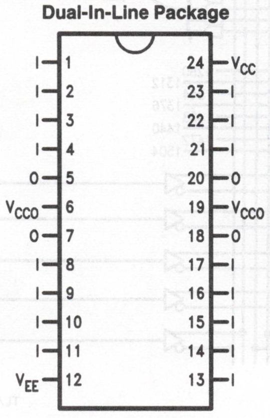

-0.9V
INPUTS
-1.7V
50%
t_{PD}
OUTPUTS
50%

TL/L/9138-7


Output Load
Dout
R_L
C_L
-2 V

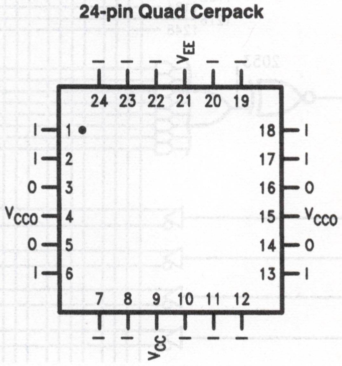
TL/L/9138-2

Connection Diagrams


Dual-In-Line Package
I 1
I 2
I 3
I 4
O 5
V_{CCO} 6
O 7
I 8
I 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 I
20 O
19 V_{CCO}
18 O
17 I
16 I
15 I
14 I
13 I

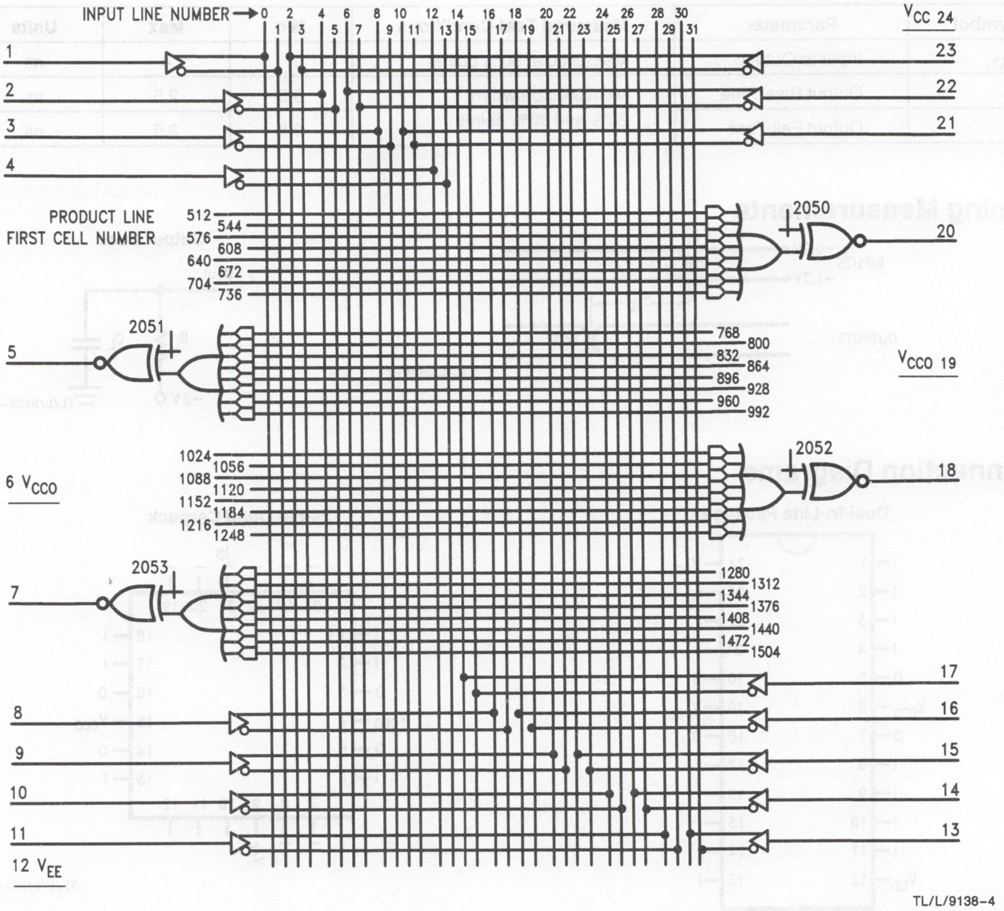
TL/L/9138-3

Top View


24-pin Quad Cerpack
V_{EE}
24 23 22 21 20 19
I 1
I 2
O 3
V_{CCO} 4
O 5
I 6
18 I
17 I
16 O
15 V_{CCO}
14 O
13 I
7 8 9 10 11 12
V_{CC}

TL/L/9138-5

Logic Diagram PAL1016P4A/PAL10016P4A


INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
VCC 24
1
2
3
4
23
22
21
PRODUCT LINE FIRST CELL NUMBER
512
544
576
608
640
672
704
736
2050
20
2051
5
768
800
832
864
896
928
960
992
VCCO 19
1024
1056
1088
1120
1152
1184
1216
1248
6 VCCO
2052
18
2053
7
1280
1312
1344
1376
1408
1440
1472
1504
17
16
15
14
13
8
9
10
11
12 VEE

TL/L/9138-4

JEDEC logic array cell number = product line first cell number + input line number

Functional Testing

As with all field-programmable devices, the user of the ECL PAL devices provides the final manufacturing step. While National's PAL devices undergo extensive testing when they are manufactured, their logic function can be fully tested only after they have been programmed to the user's pattern.

To ensure that the programmed PAL devices will operate properly in your system, National Semiconductor (along with most other manufacturers of PAL devices) strongly recommends that devices be functionally tested before being installed in your system. Even though the number of post-programming functional failures is small, testing the logic function of the PAL devices before they reach system assembly will save board debugging and rework costs. Refer to National Semiconductor's Application Note #351 and the *Programmable Logic Design Guide* for more information about the functional testing of PAL devices.

Please contact your local sales office for a list of current programming support tools for ECL PAL devices.

National Semiconductor

PRELIMINARY

PAL10/10012C4A
4 ns ECL Programmable Array Logic

General Description

The PAL10/10012C4A is a member of the National Semiconductor ECL PAL® family. The ECL PAL Series-A is characterized by 4 ns maximum propagation delays. The pinout, JEDEC fuse-map format and programming algorithm of these devices are compatible with those of all prior ECL PAL products from National. Series-A ECL PAL devices are manufactured using National Semiconductor's advanced oxide-isolated process with proven titanium-tungsten fuse technology to provide high-speed user-programmable replacements for conventional ECL SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming fuse links to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products.

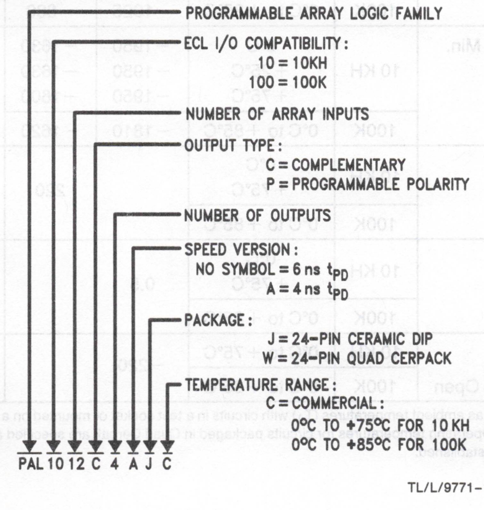
The PAL10/10012C4A logic array has a total of 12 complementary input pairs, 32 product terms and 4 complementary output functions. Each output function is the OR-sum of 8 product terms. Each product term is satisfied when all array inputs which are connected to it (via intact fuses) are in the desired state. Complementary outputs eliminate the need for external inverters and allow for more convenient output OR-tying. They are also suitable for differential sensing for increased noise immunity. All input pins have on-chip 50 kΩ pull-down resistors.

Programming equipment and software make PAL design development quick and easy. Programming is accomplished using TTL voltage levels and is therefore supported by several conventional TTL PLD programming units. After programming and verifying the logic array, an additional security fuse may be programmed to prevent direct copying of proprietary logic designs.

Features

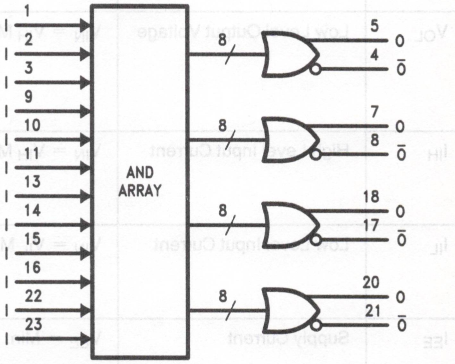
- High speed: t_{PD} = 4 ns max
- Programmable replacement for ECL logic
- Both 10KH and 100K I/O compatible versions
- Four output functions with complementary outputs
- Reliable titanium-tungsten fuses
- Security fuse to prevent direct copying
- Programmed on conventional TTL PLD programmers
- Fully Supported by PLAN™ Software
- Packaging:
  - 24-pin thin DIP (0.300″)
  - 24-pin Quad Cerpack

Ordering Information


PROGRAMMABLE ARRAY LOGIC FAMILY
ECL I/O COMPATIBILITY:
10 = 10KH
100 = 100K
NUMBER OF ARRAY INPUTS
OUTPUT TYPE:
C = COMPLEMENTARY
P = PROGRAMMABLE POLARITY
NUMBER OF OUTPUTS
SPEED VERSION:
NO SYMBOL = 6 ns tPD
A = 4 ns tPD
PACKAGE:
J = 24-PIN CERAMIC DIP
W = 24-PIN QUAD CERPACK
TEMPERATURE RANGE:
C = COMMERCIAL:
0°C TO +75°C FOR 10KH
0°C TO +85°C FOR 100K
PAL 10 12 C 4 A J C

TL/L/9771-1

Block Diagram PAL10/10012C4A


I 1
I 2
I 3
I 9
I 10
I 11
I 13
I 14
I 15
I 16
I 22
I 23
AND ARRAY
8
5 O
4 Ō
7 O
8 Ō
18 O
17 Ō
20 O
21 Ō

TL/L/9771-2

V_{EE} = 12, V_{CC} = 24, V_{CCO} (4, 5, 7, 8) = 6, V_{CCO} (17, 18, 20, 21) = 19

Pinout applies to 24-pin DIP.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
V_{EE} Relative to V_{CC}	−7V to +0.5V
Input Voltage	V_{EE} to +0.5V
Output Current	−50 mA
Lead Temperature (Soldering, 10 seconds)	300°C
ESD Tolerance	1000V

C_{ZAP} = 100 pF
R_{ZAP} = 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Units
V_{EE}	Supply Voltage	10 KH	−5.46	−5.2	−4.94	V
		100K	−4.73	−4.5	−4.27	
T	Operating Temperature (Note)	10 KH	0		+75	°C
		100K	0		+85	

Electrical Characteristics Over Recommended Operating Conditions

Output Load = 50Ω to −2.0V

Symbol	Parameter	Conditions		T_A	Min	Max	Units
V_{IH}	High Level Input Voltage	Guaranteed Input Voltage High For All Inputs	10 KH	0°C	−1170	−840	mV
				+25°C	−1130	−810	
				+75°C	−1070	−735	
			100K	0°C to +85°C	−1165	−880	
V_{IL}	Low Level Input Voltage	Guaranteed Input Voltage Low For All Inputs	10 KH	0°C	−1950	−1480	mV
				+25°C	−1950	−1480	
				+75°C	−1950	−1450	
			100K	0°C to +85°C	−1810	−1475	
V_{OH}	High Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 KH	0°C	−1020	−840	mV
				+25°C	−980	−810	
				+75°C	−920	−735	
			100K	0°C to +85°C	−1025	−880	
V_{OL}	Low Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 KH	0°C	−1950	−1630	mV
				+25°C	−1950	−1630	
				+75°C	−1950	−1600	
			100K	0°C to +85°C	−1810	−1620	
I_{IH}	High Level Input Current	$V_{IN} = V_{IH}$ Max.	10 KH	0°C		220	μA
				+75°C			
			100K	0°C to +85°C			
I_{IL}	Low Level Input Current	$V_{IN} = V_{IL}$ Min.	10 KH	0°C	0.5		μA
				+75°C			
			100K	0°C to +85°C			
I_{EE}	Supply Current	V_{EE} = Min.	10 KH	0°C to +75°C	−220		mA
		All Inputs and Outputs Open	100K	0°C to +85°C			

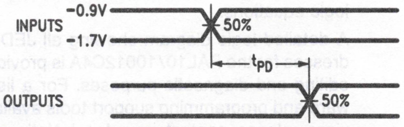
Note: Operating temperatures for circuits in Dual-In-Line packages are specified as ambient temperatures (T_A) with circuits in a test socket or mounted on a printed circuit board and transverse air flow greater than 500 linear fpm is maintained. Operating temperatures for circuits packaged in Quad Cerpak are specified as case temperatures (T_C). All specifications apply after thermal equilibrium has been established.

Switching Characteristics

Over Recommended Operating Conditions, Output load: $R_L = 50\Omega$ to $-2.0V$, $C_L = 5$ pF to GND

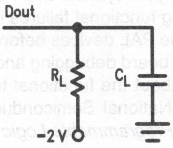
Symbol	Parameter	Measured Test Conditions	Min	Max	Units
t_{PD}	Input to Output	Measured at 50% points		4	ns
t_r	Output Rise Time	Measured between 20% and 80% points	0.5	2.5	ns
t_f	Output Fall Time		0.5	2.5	ns

Timing Measurements


-0.9V
INPUTS
-1.7V
50%
tPD
OUTPUTS
50%

TL/L/9771-3

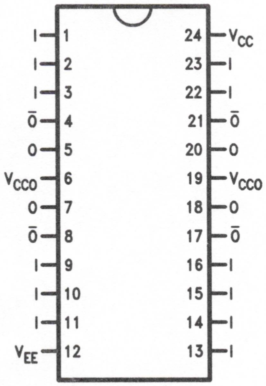
Test Load


Dout
RL
CL
-2 V

TL/L/9771-4

Connection Diagrams

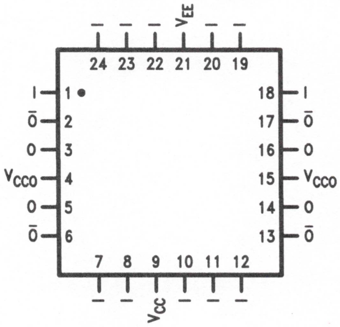
Dual-In-Line Package


I 1
I 2
I 3
Ō 4
O 5
VCCO 6
O 7
Ō 8
I 9
I 10
I 11
VEE 12
24 VCC
23 I
22 I
21 Ō
20 O
19 VCCO
18 O
17 Ō
16 I
15 I
14 I
13 I

TL/L/9771-5

Top View

24-pin Quad Cerpack


I I I VEE I I
24 23 22 21 20 19
I 1
Ō 2
O 3
VCCO 4
O 5
Ō 6
18 I
17 Ō
16 O
15 VCCO
14 O
13 Ō
7 8 9 10 11 12
I I VCC I I I

TL/L/9771-6

Functional Testing

As with all field-programmable devices, the user of the ECL PAL devices provides the final manufacturing step. While National's PAL devices undergo extensive testing when they are manufactured, their logic function can be fully tested only after they have been programmed to the user's pattern.

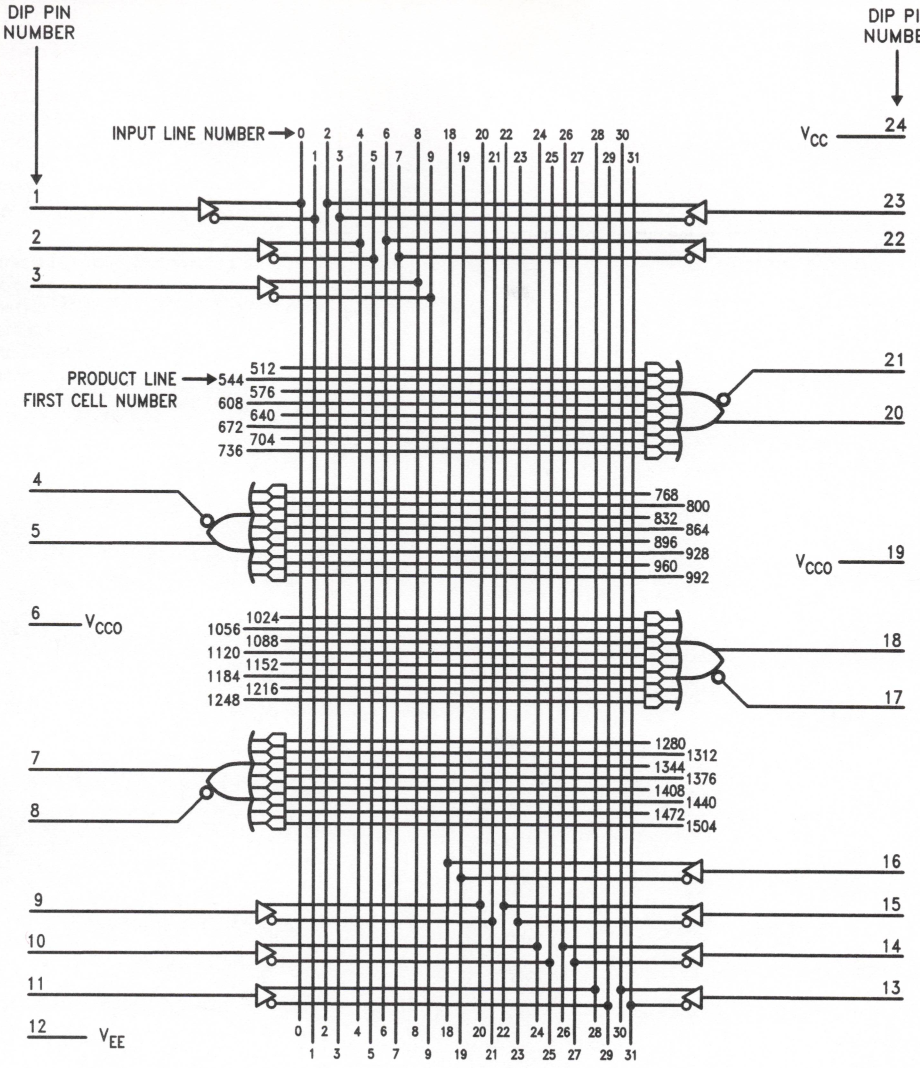
To ensure that the programmed PAL devices will operate properly in your system, National Semiconductor (along with most other manufacturers of PAL devices) strongly recommends that devices be functionally tested before being installed in your system. Even though the number of post-programming functional failures is small, testing the logic function of the PAL devices before they reach system assembly will save board debugging and rework costs. For more information about the functional testing of PAL devices, please refer to National Semiconductor's Application Note #351 and the *Programmable Logic Design Guide.*

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be downloaded into a large variety of programming equipment. Many software packages and programming units support a large variety of programmable logic products as well. The PLAN software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

A detailed logic diagram showing all JEDEC fuse-map addresses for the PAL10/10012C4A is provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the ECL PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

Logic Diagram—PAL1012C4A/PAL10012C4A


DIP PIN NUMBER
DIP PIN NUMBER
INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE FIRST CELL NUMBER
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1
2
3
4
5
6 V_{CCO}
7
8
9
10
11
12 V_{EE}
V_{CC} 24
23
22
21
20
V_{CCO} 19
18
17
16
15
14
13

TL/L/9771-7

JEDEC logic array cell number = product line first cell number + input line number

National Semiconductor

PRELIMINARY

PAL10/10016RM4A, PAL10/10016LM4A ECL Programmable Array Logic

General Description

The PAL10/10016RM4A and PAL10/10016LM4A are members of the National Semiconductor ECL PAL® family. The ECL PAL Series-A is characterized by 4 ns maximum propagation delays (combinatorial input-to-output). The pin-out, JEDEC fuse-map format and programming algorithm of these devices are compatible with those of all prior ECL PAL products from National. Series-A ECL PAL devices are manufactured using National Semiconductor's advanced oxide-isolated process with proven titanium-tungsten fuse technology to provide high-speed user-programmable replacements for conventional ECL SSI/MSI logic with significant chip-count reduction.

Programmable logic devices provide convenient solutions for a wide variety of application-specific functions, including random logic, custom decoders, state machines, etc. By programming fuse links to configure AND/OR gate connections, the system designer can implement custom logic as convenient sum-of-products Boolean functions. System prototyping and design iterations can be performed quickly using these off-the-shelf products.

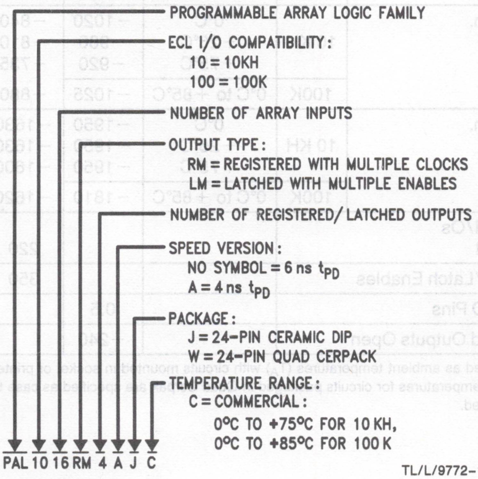
The PAL10/10016RM/LM4A logic array has a total of 16 complementary input pairs, 32 product terms and four output functions; each output function is the OR-sum of 8 product terms. The 16RM4A provides an edge-triggered D-type register on each of its four outputs. Registers allow the PAL device to implement sequential logic circuits. The 16LM4A provides D-type transparent latches on its four outputs. Polarity fuses allow each output to be active-high or active-low.

Programming equipment and software make PAL design development quick and easy. Programming is accomplished using TTL voltage levels and is therefore supported by several conventional TTL PLD programming units. After programming and verifying the logic array, an additional security fuse may be programmed to prevent direct copying of proprietary logic designs.

Features

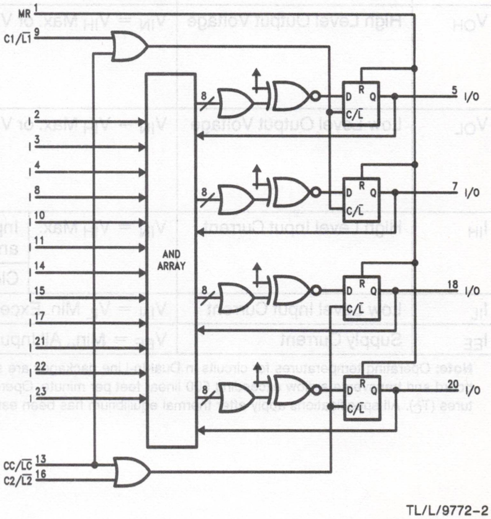
- High speed:
  - t_{SU} = 3 ns min
  - t_{CLK} = 2 ns max
  - f_{MAX} = 200 MHz max (registered)
  - t_{PD} = 4 ns max (combinatorial)
- Programmable replacement for ECL SSI/MSI logic
- Both 10 KH and 100K I/O compatible versions
- Four registered or latched output functions with I/O pin feedback; twelve dedicated inputs
- Individually programmable polarity on all logic outputs
- Reliable Titanium Tungsten fuses
- Security fuse to prevent direct copying
- Programmed on conventional TTL PLD programmers
- Fully Supported by PLAN™ Software
- Packaging:
  - 24-pin thin DIP (0.300″)
  - 24-pin Quad Cerpak

Ordering Information


PROGRAMMABLE ARRAY LOGIC FAMILY
ECL I/O COMPATIBILITY:
10 = 10KH
100 = 100K
NUMBER OF ARRAY INPUTS
OUTPUT TYPE:
RM = REGISTERED WITH MULTIPLE CLOCKS
LM = LATCHED WITH MULTIPLE ENABLES
NUMBER OF REGISTERED/LATCHED OUTPUTS
SPEED VERSION:
NO SYMBOL = 6 ns tPD
A = 4 ns tPD
PACKAGE:
J = 24-PIN CERAMIC DIP
W = 24-PIN QUAD CERPACK
TEMPERATURE RANGE:
C = COMMERCIAL:
0°C TO +75°C FOR 10 KH,
0°C TO +85°C FOR 100 K
PAL 10 16 RM 4 A J C

TL/L/9772-1

Block Diagram PAL10/10016RM/LM4A


MR 1
C1/L̄1 9
2
3
4
8
10
11
14
15
17
21
22
23
CC/L̄C 13
C2/L̄2 16
AND ARRAY
8
D R Q
C/L̄
5 I/O
7 I/O
18 I/O
20 I/O

TL/L/9772-2

V_{EE} = 12, V_{CC} = 24, V_{CCO} (5,7) = 6, V_{CCO} (18,20) = 19

Pinout applies to 24-pin DIP.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias	−55°C to +125°C
Storage Temperature Range	−65°C to +150°C
V_{EE} Relative to V_{CC}	−7V to +0.5V
Input Voltage	V_{EE} to +0.5V
Output Current	−50 mA
Lead Temperature (Soldering, 10 seconds)	300°C
ESD Tolerance	1000V

C_{ZAP} = 100 pF
R_{ZAP} 1500Ω
Test Method: Human Body Model
Test Specification: NSC SOP-5-028

Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Units
V_{EE}	Supply Voltage	10 KH	−5.46	−5.2	−4.94	V
		100K	−4.73	−4.5	−4.27	
T	Operating Temperature (Note)	10 KH	0		+75	°C
		100K	0		+85	
R_L	Standard 10 KH/100K Load			50		Ω
C_L	Standard 10 KH/100K Load			5		pF
t_{SU}	Setup Time of Input or Feedback		3.0			ns
t_H	Input Hold Time		0			ns
t_W	Clock or Enable Pulse Width		2.0			ns
t_{WMR}	Master Reset Pulse Width		2.0			ns

Electrical Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Conditions		T_A	Min	Max	Units
V_{IH}	High Level Input Voltage	Guaranteed Input Voltage High For All Inputs	10 KH	0°C	−1170	−840	mV
				+25°C	−1130	−810	
				+75°C	−1070	−735	
			100K	0°C to +85°C	−1165	−880	
V_{IL}	Low Level Input Voltage	Guaranteed Input Voltage Low For All Inputs	10 KH	0°C	−1950	−1480	mV
				+25°C	−1950	−1480	
				+75°C	−1950	−1450	
			100K	0°C to +85°C	−1810	−1475	
V_{OH}	High Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 KH	0°C	−1020	−840	mV
				+25°C	−980	−810	
				+75°C	−920	−735	
			100K	0°C to +85°C	−1025	−880	
V_{OL}	Low Level Output Voltage	$V_{IN} = V_{IH}$ Max. or V_{IL} Min.	10 KH	0°C	−1950	−1630	mV
				+25°C	−1950	−1630	
				+75°C	−1950	−1600	
			100K	0°C to +85°C	−1810	−1620	
I_{IH}	High Level Input Current	$V_{IN} = V_{IH}$ Max.	Inputs, I/Os and MR			220	µA
			Clocks/Latch Enables			350	
I_{IL}	Low Level Input Current	$V_{IN} = V_{IL}$ Min. Except I/O Pins			0.5		µA
I_{EE}	Supply Current	V_{EE} = Min., All Inputs and Outputs Open			−240		mA

Note: Operating temperatures for circuits in Dual-In-Line packages are specified as ambient temperatures (T_A) with circuits mounted in socket or printed circuit board and transverse airflow exceeding 500 linear feet per minute. Operating temperatures for circuits packaged in Quad Cerpak are specified as case temperatures (T_C). All specifications apply after thermal equilibrium has been established.

Switching Characteristics Over Recommended Operating Conditions

Output Load: $R_L = 50\Omega$ to $-2.0V$, $C_L = 5$ pF to GND

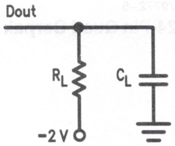
Symbol	Parameter	Measured From	Measured To	Min	Max	Units
t_{CLK} (Note 1)	Clock to Output or Feedback	C_n ↑	I/O		2.0	ns
t_{LE} (Note 2)	Enable to Output or Feedback	$\overline{L_n}$ ↓	I/O		2.0	ns
t_{PD} (Note 2)	Input or Feedback to Output	I	I/O		4.0	ns
t_{MR} (Note 1, 2)	Master Reset to Output	MR ↑	I/O ↓		3.5	ns
f_{MAX} (Note 1, 3)	Maximum Frequency				200	MHz
t_r	Output Rise Time	Measured Between 20% and 80% points		0.5	2.0	ns
t_f	Output Fall Time			0.5	2.0	ns

Note 1: Applies to 16RM4.

Note 2: Applies to 16LM4.

Note 3: $f_{MAX} = (t_{SU} + t_{CLK})^{-1}$

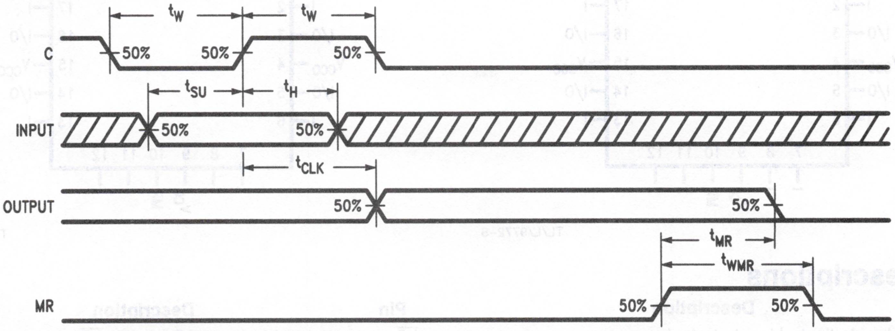
Test Load



Dout
R_L
C_L
-2 V

TL/L/9772-3

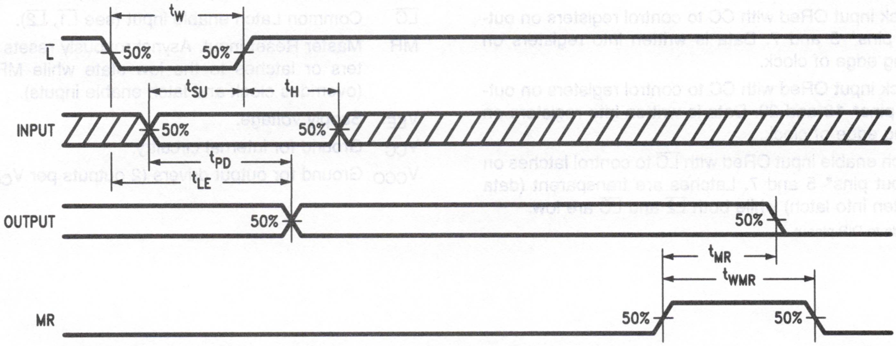
Timing Waveform PAL10/10016RM4A



C
INPUT
OUTPUT
MR
t_W
t_W
t_{SU}
t_H
t_{CLK}
t_{MR}
t_{WMR}
50%

TL/L/9772-4

Timing Waveform PAL10/10016LM4A



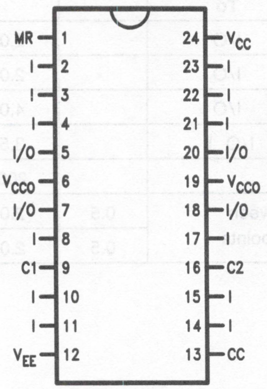
L
INPUT
OUTPUT
MR
t_W
t_{SU}
t_H
t_{PD}
t_{LE}
t_{MR}
t_{WMR}
50%

TL/L/9772-5

Connection Diagrams

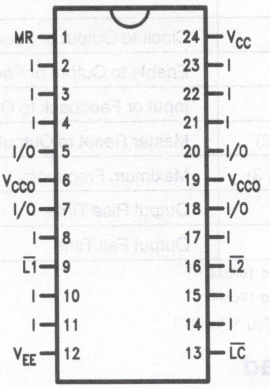
24-Pin Dual-In-Line Packages

PAL10/10016RM4A


MR 1
I 2
I 3
I 4
I/O 5
V_{CCO} 6
I/O 7
I 8
C1 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 I
20 I/O
19 V_{CCO}
18 I/O
17 I
16 C2
15 I
14 I
13 CC

TL/L/9772-6

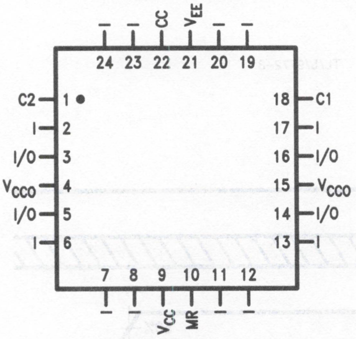
PAL10/10016LM4A


MR 1
I 2
I 3
I 4
I/O 5
V_{CCO} 6
I/O 7
I 8
$\overline{L1}$ 9
I 10
I 11
V_{EE} 12
24 V_{CC}
23 I
22 I
21 I
20 I/O
19 V_{CCO}
18 I/O
17 I
16 $\overline{L2}$
15 I
14 I
13 $\overline{LC}$

TL/L/9772-7

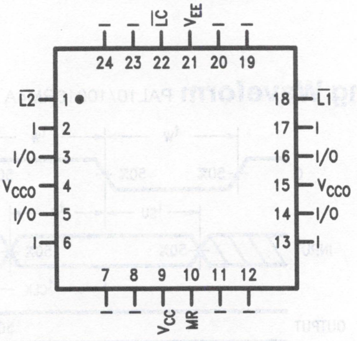
24-Pin Quad Cerpak

PAL10/10016RM4A


I 24
I 23
CC 22
V_{EE} 21
I 20
I 19
C2 1
I 2
I/O 3
V_{CCO} 4
I/O 5
I 6
18 C1
17 I
16 I/O
15 V_{CCO}
14 I/O
13 I
7 I
8 I
9 V_{CC}
10 MR
11 I
12 I

TL/L/9772-8

PAL10/10016LM4A


I 24
I 23
$\overline{LC}$ 22
V_{EE} 21
I 20
I 19
$\overline{L2}$ 1
I 2
I/O 3
V_{CCO} 4
I/O 5
I 6
18 $\overline{L1}$
17 I
16 I/O
15 V_{CCO}
14 I/O
13 I
7 I
8 I
9 V_{CC}
10 MR
11 I
12 I

TL/L/9772-9

Pin Descriptions

Pin	Description
I	Twelve dedicated inputs to logic array.
I/O	Four outputs from registered or latched logic functions. Also provides feedback and may be used for additional inputs to logic array.
C1	Clock input ORed with CC to control registers on output pins* 5 and 7. Data is written into registers on rising edge of clock.
C2	Clock input ORed with CC to control registers on output pins* 18 and 20. Data is written into registers on rising edge of clock.
$\overline{L1}$	Latch enable input ORed with $\overline{LC}$ to control latches on output pins* 5 and 7. Latches are transparent (data written into latch) while both $\overline{L2}$ and $\overline{LC}$ are low.
$\overline{L2}$	Latch enable input ORed with $\overline{LC}$ to control latches on output pins* 18 and 20. Latches are transparent (data written into latch) while both $\overline{L2}$ and $\overline{LC}$ are low.
CC	Common Clock input (see C1, C2).
$\overline{LC}$	Common Latch enable input (see $\overline{L1}$, $\overline{L2}$).
MR	Master Reset input. Asynchronously resets all registers or latches to the low state while MR is high (overrides clock and latch enable inputs).
V_{EE}	Supply voltage.
V_{CC}	Ground for internal circuitry.
V_{CCO}	Ground for output drivers (2 outputs per V_{CCO}).

*Corresponds to DIP pinout

Functional Description

The PAL10/10016RM/LM4A consists of a single programmable AND-gate array with fixed OR-gate connections. The AND array consists of 16 complementary pairs of input lines crossing 32 product-term lines with a programmable fuse at each intersection (1024 fuses). The product terms are organized into four groups of eight each. The eight product terms in each group are connected into an OR-gate to produce the sum-of-products logic function.

An unprogrammed fuse establishes a connection between an input line (true or complement phase of an array input signal) and a product term. Programming the fuse removes the connection. A product term is satisfied (logically true) while all the input lines connected to it (via intact fuses) are in the proper logic state. Therefore, if both the true and complement of at least one array input are left connected to a product line, that product term would always be held in the low logic state (which is the state of all product terms in an unprogrammed device).

The four outputs of the PAL10/10016RM4A pass through D-type registers triggered on the high-going edge of the appropriate clock input. The outputs of the PAL10/10016LM4A pass through D-type transparent latches which are enabled (pass data) while the appropriate enable inputs are low. The four registers or latches are separated into two pairs. A separate clock/latch-enable input is provided for each pair. An additional common clock/enable input is ORed with each (see logic diagrams which follow).

The AND-OR logic functions can be optionally inverted before the registers or latches. Polarity inversion is controlled by an individual "polarity fuse" associated with each output function (the original unprogrammed state produces active-high logic functions). Device output pins always indicate active-high register/latch outputs.

The I/O pins used for outputting the registered/latched logic functions also feed back into the logic array as additional inputs. This is useful, for example, to implement sequential circuits with registered parts. Any of these I/O pins may, instead, be used as an additional dedicated input pin. By leaving the associated logic function unprogrammed, the output driver would remain in the low logic state allowing an externally-applied signal to control the array input.

Logic functions requiring more than eight product terms can be implemented conveniently by OR-tying two (or more) device outputs. Partial sums are formed on each of the OR-tied output functions. Each function, however, must be programmed for active-high output polarity, and the associated registers/latches should be controlled by the same clock/enable signal. Each of the array inputs fed back from the OR-tied I/O pins would indicate the correct final logic function.

All input and I/O pins have on-chip 50 kΩ pull-down resistors.

Functional Testing

As with all field-programmable devices, the user of ECL PAL devices provides the final manufacturing step. While National's PAL devices undergo extensive testing when they are manufactured, their logic function can be fully tested only after they have been programmed to the user's pattern.

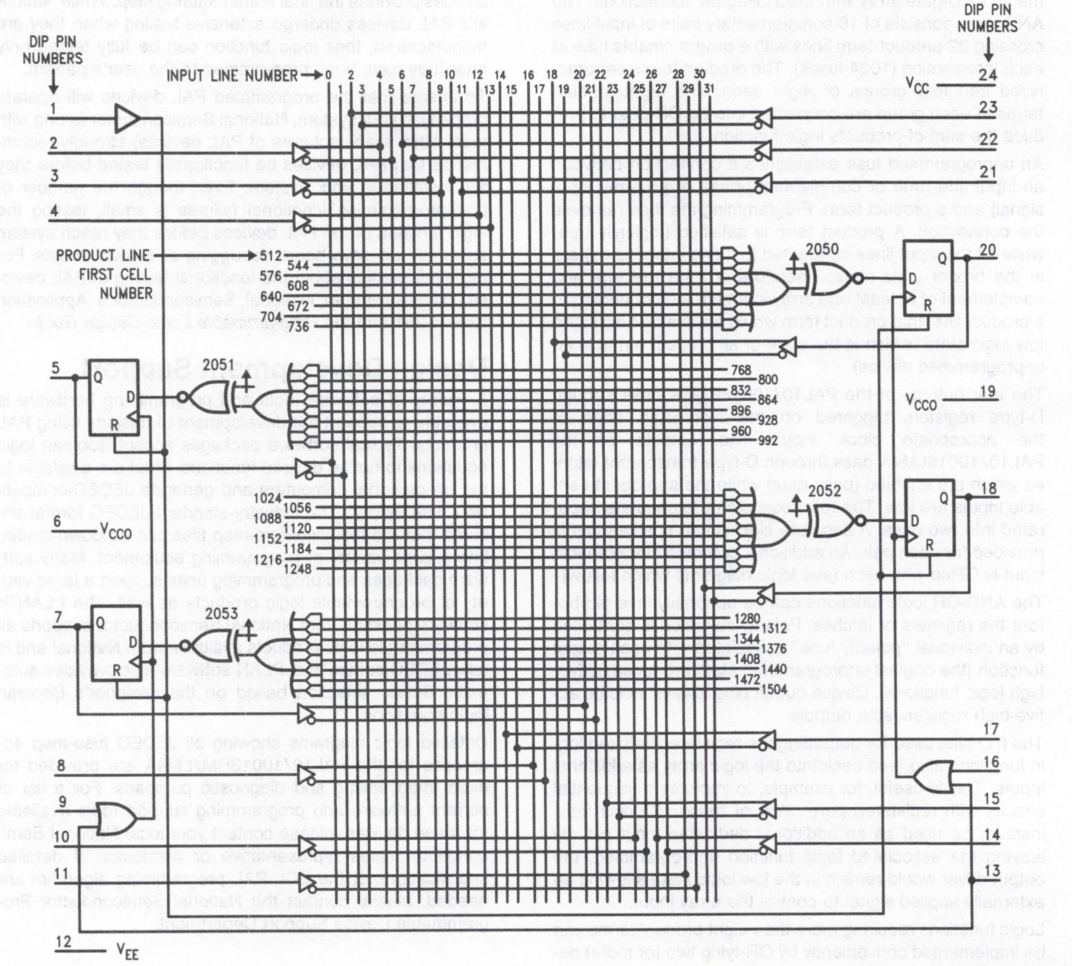
To ensure that the programmed PAL devices will operate properly in your system, National Semiconductor (along with most other manufacturers of PAL devices) strongly recommends that PAL devices be functionally tested before they are installed in your system. Even though the number of post-programming functional failures is small, testing the logic function of the PAL devices before they reach system assembly will save board debugging and rework costs. For more information about the functional testing of PAL devices, please refer to National Semiconductor's Application Note #351 and the *Programmable Logic Design Guide.*

Design Development Support

A variety of software tools and programming hardware is available to support the development of designs using PAL products. Typical software packages accept Boolean logic equations to define desired functions. Most are available to run on personal computers and generate JEDEC-compatible "fuse maps". The industry-standard JEDEC format ensures that the resulting fuse-map files can be down-loaded into a large variety of programming equipment. Many software packages and programming units support a large variety of programmable logic products as well. The PLAN™ software package from National Semiconductor supports all programmable logic products available from National and is fully JEDEC-compatible. PLAN software also provides automatic device selection based on the designer's Boolean logic equations.

Detailed logic diagrams showing all JEDEC fuse-map addresses for the PAL10/10016RM/LM4A are provided for direct map editing and diagnostic purposes. For a list of current software and programming support tools available for these devices, please contact your local National Semiconductor sales representative or distributor. If detailed specifications of the ECL PAL programming algorithm are needed, please contact the National Semiconductor Programmable Device Support Department.

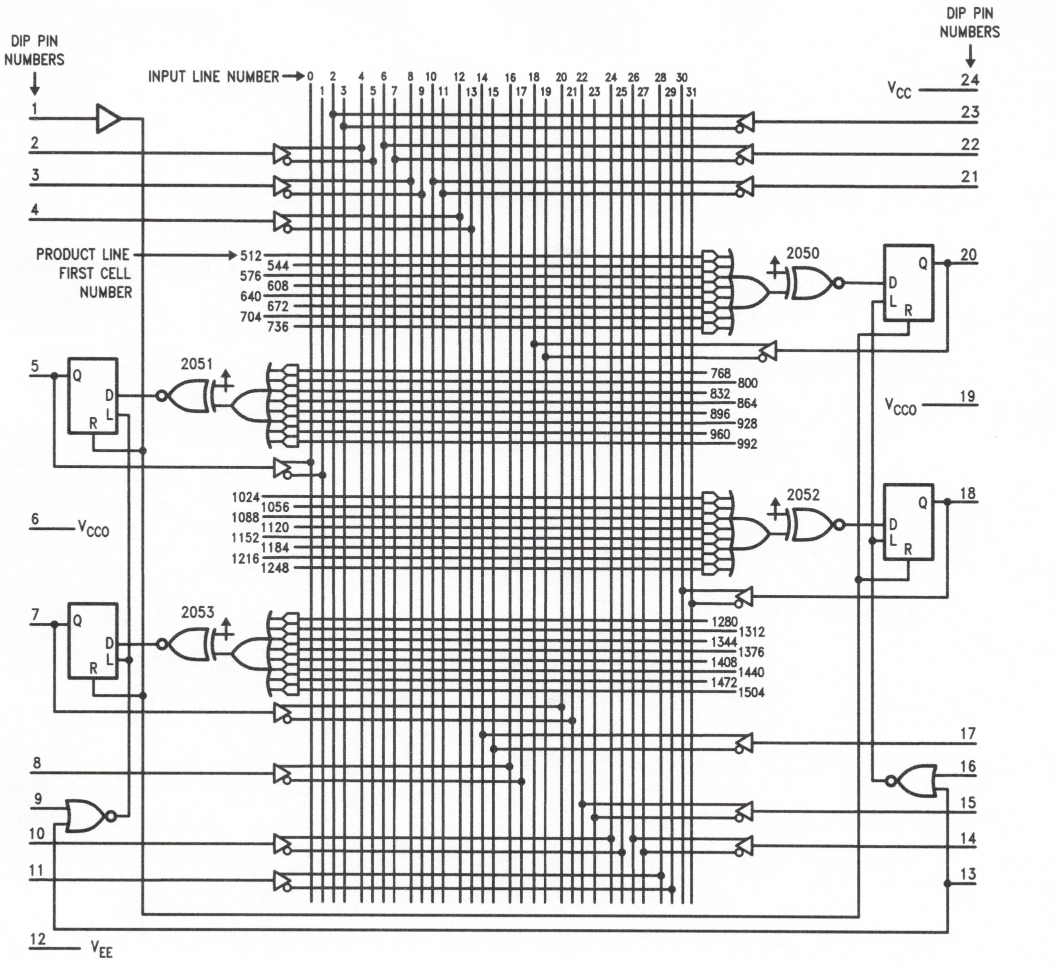
Logic Diagram—PAL1016RM4A/PAL10016RM4A


DIP PIN NUMBERS
INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
DIP PIN NUMBERS
24
V_{CC}
23
22
21
1
2
3
4
PRODUCT LINE FIRST CELL NUMBER
512
544
576
608
640
672
704
736
2050
20
Q
D
R
5
Q
D
R
2051
768
800
832
864
896
928
960
992
19
V_{CCO}
1024
1056
1088
1120
1152
1184
1216
1248
6
V_{CCO}
2052
18
Q
D
R
7
Q
D
R
2053
1280
1312
1344
1376
1408
1440
1472
1504
17
16
15
14
13
8
9
10
11
12
V_{EE}

TL/L/9772-10

JEDEC logic array cell number = product line first cell number + input line number.

Logic Diagram—PAL1016LM4A/PAL10016LM4A


DIP PIN NUMBERS
DIP PIN NUMBERS
INPUT LINE NUMBER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
PRODUCT LINE FIRST CELL NUMBER
512
544
576
608
640
672
704
736
768
800
832
864
896
928
960
992
1024
1056
1088
1120
1152
1184
1216
1248
1280
1312
1344
1376
1408
1440
1472
1504
2050
2051
2052
2053
Q
D
L
R
1
2
3
4
5
6
V_{CCO}
7
8
9
10
11
12
V_{EE}
V_{CC}
24
23
22
21
20
19
V_{CCO}
18
17
16
15
14
13

TL/L/9772–11

JEDEC logic array cell number = product line first cell number + input line number.

PART II
DESIGN GUIDE

Designing with PLDs 3

PLD Design Methodology 4

PLD Design Development Tools 5

Fabrication of Programmable Logic 6

Application Examples 7

Appendices/Physical Dimensions 8

Section 3

Designing with Programmable Logic

Section 3 Contents

3.1 Background to PLDs 3-3
  PROM devices
  FPLA devices
  PAL devices
  GAL devices
3.2 Design Advantages 3-3
  Simplified system design
  Increased functional density
3.3 Manufacturing Advantages 3-5
  Cost of inventory
  Cost of ownership
  Cost of upgrades and modifications
  Higher reliability
3.4 Alternative Methodologies 3-6
  Standard LSI
  Standard SSI/MSI
  Full custom
  Semi-custom ASICs
3.5 PLD Architecture Overview 3-6
  Logic array structure
  Registered outputs
  Exclusive-OR gates
  Programmable output polarity
  Advanced product features
3.6 Programmable Logic from National 3-11
  TTL PROMs
  Lateral fuse TTL PAL products
  Vertical fuse TTL PAL products
  ECL PAL products
  E^2CMOS GAL products
  Pre-programmed and mask-programmed products
3.7 Design Development Tools 3-12
  Programming hardware
  Development software
  NSC PLAN software
  Other assemblers/compilers

National Semiconductor

Section 3
Designing with Programmable Logic

Programmable logic has evolved over the last decade into a design tool permitting digital logic designs with a minimal number of packages and a maximum of flexibility. The key to PLDs is the use of embedded programmable cells (typically fuse-links) which allow logic components to be configured into specific designs in the field. This permits logic consolidation with quick implementation and equally quick design revision often without board layout changes.

While Programmable Logic Devices (PLDs) do not offer the density of standard VLSI or custom circuits, they are far more flexible than the former and more cost-efficient than the latter. They have found extensive use in varied applications. They are both inexpensive and space-saving in replacing less efficient "glue logic", which was one of the more popular original uses of PLDs. But they are also capable of efficiently implementing complex functions and state machines.

3.1 Background to PLDs

The use of programmable logic in digital design began with the diode matrix with aluminum fuses at the crosspoints in the early 1960's. This evolved into the PROM through the addition of a decoder at the inputs. The result was an addressable memory which could also be seen as a universal logic device with a fixed AND matrix (the decoder) feeding a programmable OR matrix (the diode array). A representative PROM circuit is shown in *Figure 3-1(b)* as implementing, for example, a simple set of equations given in *Figure 3-1(a)*. The disadvantage of a PROM used as a logic device derives from its universality. The number of product terms available is 2^n, where n is the number of input variables. Each additional variable (input pin) doubles the size of the matrix. As a result, commercial PROMs offer limited input (i.e. address pins). This approach rapidly degrades performance due to the decode logic and the array dimensions required, and increases cost through inefficient use of silicon. Many logic applications require more inputs, but not the flexibility of a full decoder.

This dilemma was solved by introducing a second fuse matrix in place of the fixed decoder, allowing selection only of those product terms required by the design. This made much more efficient use of the programmable matrix. Like the PROM, it was made using fuses which could be configured in the field and was therefore called the Field Programmable Logic Array (FPLA or just PLA). The basic PLA architecture is shown in *Figure 3-1(c)*. Unlike the PROM, the PLA can handle logic functions requiring more input variables with much less than 2^n product terms.

However, the additional fuses of the second matrix in the PLA require additional selection and programming circuitry, which makes it intrinsically more expensive than a comparable PROM. This can lead to cost inefficiencies when only a few product terms are being used. Also, a programmable array imposes a longer delay than a hardwired decoder and is the dominant factor in signal delay through a PLD. Therefore, it is difficult for a PLA with programmable AND and OR arrays to compete with the speed of a PROM of similar size.

Cost efficiencies comparable to the PROM were achievable only by reducing the overhead circuitry necessary. One solution was to hardwire the OR array and allow the user to program only the AND array. This arrangement is known as the Programmable Array Logic (PAL®) architecture shown in *Figure 3-1(d)*. The introduction of the PAL device was the key which unlocked the potential of efficient programmable logic for designers. The PAL device could be made more cost-effectively than the PLA and could substitute flexibility in the OR array by being offered in a variety of basic configurations. Also, since the number of programmable arrays through which a logic signal needed to pass was reduced from two to one, device performance improved considerably.

Development of the initial PAL concept has led to families of products in several technologies, offering a range of design building blocks, power requirements and performance. Developments in cell technology beyond the original metal fuse-links have led to the "vertical fuse" programming cell, offering higher programming yields and faster signal propagation.

The advantages of the one-time programmable (OTP) devices described above hinge on the ability to configure integrated circuits in the field. Once blown, the cells cannot be reconfigured. More cost savings would be available if PLDs could be reconfigured. This would permit device reuse and exhaustive factory testing for yet higher programming yield and improved reliability. Recent developments in semiconductor technology have made electrically erasable cells available for memory and logic products. Such reconfigurable cells have been used to make "Generic Array Logic" (GAL®) devices. Basic GAL devices offer not only all the logic configurations likely to be required but also allow modification of prototypes for development debug and also of systems in the field for reconfiguration or upgrade.

3.2 Design Advantages

Digital logic designers have always worked under constraints. Reduction of system size and cost demand efficient, compact designs. System reliability forces designers to compromise between evolving solutions and existing proven methods. Future revisions demand a degree of flexibility which must be anticipated. Yet the systems themselves increase in complexity, components sophistication requires ever more sophisticated tools and conceiving the optimal design for so many parameters requires a range of skills which must constantly be developed.

LOGIC EQUATIONS

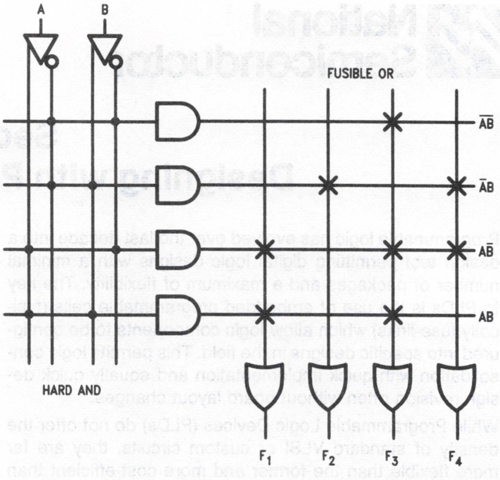
$F_1 = A$

$F_2 = \overline{A}B$

$F_3 = A + \overline{B}$

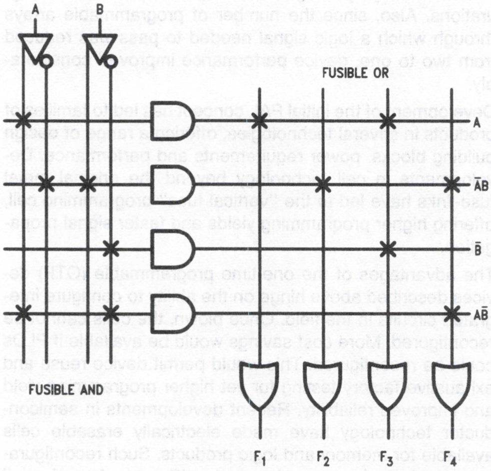
$F_4 = \overline{A}B + A\overline{B}$

(a) Desired Logic Functions


A
B
FUSIBLE OR
$\overline{A}\overline{B}$
$\overline{A}B$
$A\overline{B}$
AB
HARD AND
F_1
F_2
F_3
F_4

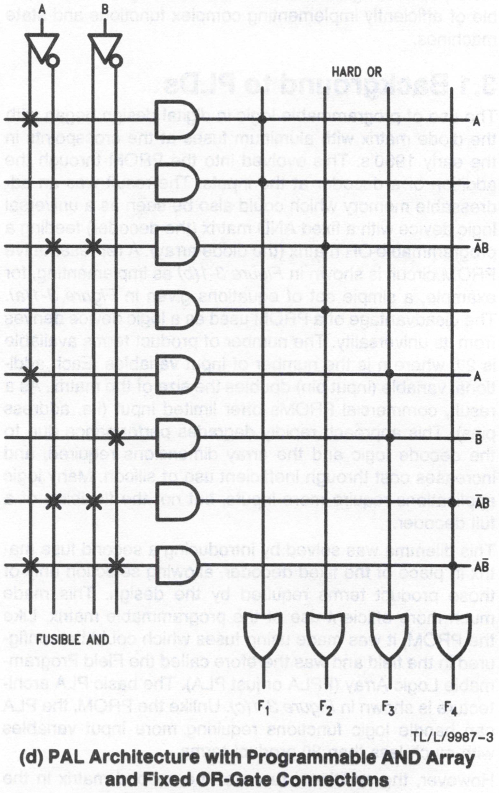
TL/L/9987-1

(b)PROM Architecture with Fixed AND (Decoder) and Programmable OR Array


A
B
FUSIBLE OR
A
$\overline{A}B$
$\overline{B}$
$A\overline{B}$
FUSIBLE AND
F_1
F_2
F_3
F_4

TL/L/9987-2

(c) PLA Architecture with Programmable AND & OR Arrays


A
B
HARD OR
A
$\overline{A}B$
A
$\overline{B}$
$\overline{A}B$
$A\overline{B}$
FUSIBLE AND
F_1
F_2
F_3
F_4

TL/L/9987-3

(d) PAL Architecture with Programmable AND Array and Fixed OR-Gate Connections

FIGURE 3-1. Comparison of Programmable Logic Basic Architectures

PLDs do not solve all of these problems. But they do provide a method of dealing with some of the major issues in an effective way by providing a uniform methodology. This section discusses some of the major advantages available to designers through the use of PLDs.

SIMPLIFIED SYSTEM DESIGN

The semi-custom approach of PLDs allows the user to specify exactly the functions which will be implemented in the logic. This avoids the problem of interconnecting various SSI components to achieve the same result. At the same time, PLDs offer speed advantages through reduction of interconnects. The methodology becomes one of writing the equations for the desired function with the help of the software tools and using such equations to configure the appropriate devices. This methodology accelerates both the conception and implementation of the design. Since the software tools available handle all the details regarding device configuration, the designer is left free to focus on the design of the application itself. An additional benefit is that many of the changes which usually need to be incorporated in a design after implementation can often be accommodated by altering the PLD's internal configuration, thus avoiding rewiring of prototype and printed circuit boards.

INCREASED FUNCTIONAL DENSITY

Despite the development of LSI and VLSI devices which package an amazing amount of logic on a single chip, system designers have still had to contend with the power, space and drive problems associated with a myriad of SSI/MSI packages used either for "glue" or for specific design requirements not available in off-the-shelf parts. Ordinarily, this will decrease the functional density.

PLDs, however, offer a compact solution with high functionality and less waste in I/O and interconnect lines, so that functional density can approach that of custom logic without the associated engineering charges. On the other hand, the combination of several functions on a single chip reduces power as well as space and has the added benefit of boosting system performance through a reduction of interconnects.

3.3 Manufacturing Advantages

While PLDs offer a number of advantages over SSI/MSI for the designer, there are a number of considerations which only become apparent when system volume production is examined. These include:

- Cost of Inventory
- Cost of Ownership
- Cost of Upgrades/Modifications
- Reliability

COST OF INVENTORY

A hidden cost associated with many designs is the inventory of parts required in order to sustain it in production. PLDs are able to reduce this cost by offering a substantial reduction in the number of different part types which are otherwise required to build a given system using standard logic parts. This is particularly true for GAL devices. Users may find that the inventory cost advantages of GAL devices tend to offset the slight difference in price of GAL over standard PAL devices.

COST OF OWNERSHIP

The cost of ownership of a particular part is more subtle than the simple price at which it is available on the market. In fact, the cost of ownership includes the cost of those devices which fail and which must be replaced. This cost increases dramatically as the discovery of failures occurs later in the production process.

The additional cost of a failed part at incoming inspection is relatively minor. However, PLDs must be programmed to the user's pattern before any meaningful functional testing can be done by the user. Therefore, the first detectable device failure for a PLD will be a programming failure detected during device verification on the programming equipment. This is the most frequent of all failure modes for PLDs. Vertical-fuse PAL devices and electrically-erasable GAL devices have a much higher factory testability and an inherently more reliable programming technology. This gives these advanced technology products a strong advantage at this early stage of production.

Once devices have been verified, functional testing can be performed while still loaded in the programming unit or on production IC testers. Otherwise, the device functionality is tested in-circuit. Both involve further production costs which contribute to cost of ownership, particularly if the device is already soldered in place.

Failures beyond this may occur at the board or system level. This sometimes occurs despite testing at previous stages and happens to standard non-programmable products as much as to PLDs. Most are detected in production, but are increasingly costly to correct.

The final location at which device failure over product lifetime can occur is in the field. Field failures are attributed primarily to device reliability. Since field failures have the highest associated cost, National Semiconductor performs extensive reliability testing on all PLD products, processes and packages. Continuing developments in circuit and programming technologies are creating inherently more reliable PLDs, as exemplified by the vertical-fuse PAL devices.

COST OF UPGRADES AND MODIFICATIONS

Unlike standard SSI/MSI, PLDs offer some degree of flexibility in permitting alterations to a circuit which is already in production. At its simplest level, all PLDs permit some degree of reconfiguration within the existing printed circuit board and device pinout. Even if the original one-time-programmable (OTP) PLD cannot be reconfigured, subsequent production can alter the circuit by altering the fuse map without any other changes. Where the change is more dramatic, the use of a pin-compatible GAL device may still offer a change which requires no circuit board alterations. Field alterations are then limited to replacing a device and do not require the standard time-consuming cut-and-jumper approach.

HIGHER RELIABILITY

The higher levels of integration associated with contemporary digital design have brought reliability and testability to the fore as issues in system design. Increased system and integrated circuit complexity have made it crucial that an acceptable design debug and system test methodology be included as part of the development process. PLDs help in this by being both flexible and versatile in design debug, particularly reconfigurable GAL devices.

It has also been shown that system reliability is a function of the number of components used and the pins and wiring necessary to interconnect them. The decreased package count over standard SSI/MSI devices through the use of PLDs therefore helps maximize system reliability.

Recent advances in PLD circuit design and processing technology have greatly improved the long-term reliability of both OTP and reprogrammable E^2CMOS products. Sophisticated test circuitry built into these products allows increased testability of on-chip circuit elements and programming cells to ensure long-term reliability.

3.4 Alternative Methodologies

Since the introduction of LSI, there has been a growing "complexity gap" between high-density, high-functionality devices and the low-density device available to interconnect them or provide non-standard design alternatives. The advent of PLDs has helped somewhat to bridge that gap.

STANDARD LSI

Advances in this area have been made at the cost of device flexibility and support software complexity. A hidden disadvantage has been the need to interface such devices into a given system, often resulting in a disproportionately high package count and power supply problems. PLDs can be used for package reduction of the inevitable "glue logic" around LSI applications. But also, frequently design requirements may be for a controller which doesn't require microprocessor complexity or cannot accept the slower microprocessor speeds. While standard LSI attempts to be a universal solution, the system under design may require a much simpler solution and/or special functions not provided in available LSI. PLDs can often provide a more appropriate, higher-performance, cost-effective and compact design in such cases.

STANDARD SSI/MSI

While having mounting competition from other design methodologies, SSI/MSI logic continues to be used in many designs where specialized functions are required in lower production volumes. PLDs offer a more effective means of implementing these functions in all but the most trivial examples because of their ability to reduce package count, increase performance, offer design support tools and simplify revisions and upgrades. Where systems require absolute minimum delays through short logic paths or standard logic functions, standard SSI/MSI devices are often an indispensable solution. However, in most other "glue logic" applications, the long-term efficiency and flexibility of PLDs tend to outweigh any initial parts cost savings derived from using SSI/MSI, especially when design revision is considered.

FULL CUSTOM

These provide an excellent solution to well-defined, low-to-medium complexity logic which is expected to be produced in very high volume. The risks involved in committing both the time and money mean it is seldom used in practice unless extreme performance/density is required or extreme high volume is expected.

SEMI-CUSTOM ASIC's

The success of this semi-custom approach depends on the efficiency of use of the gates available. This requires skillful partitioning of the logic and careful selection of the gate array. It has far less development time and cost associated with it than full custom, but fixed costs must still be considered, along with the difficulty of correcting any problem in the logic once committed to silicon.

Even though prototype development time for gate arrays has been significantly reduced over the past few years, almost any redesign requirements can imply mask revision and have a devastating effect on system introduction.

3.5 PLD Architecture Overview

All of the PLDs currently offered by National are PAL-type architectures (except the GAL39V18 which will be a PLA architecture). This section describes some of the basic architectural features found in the various PAL-like devices (including the GAL devices). For more detailed descriptions of the architectures of specific devices, refer to the appropriate datasheet in Section 2.

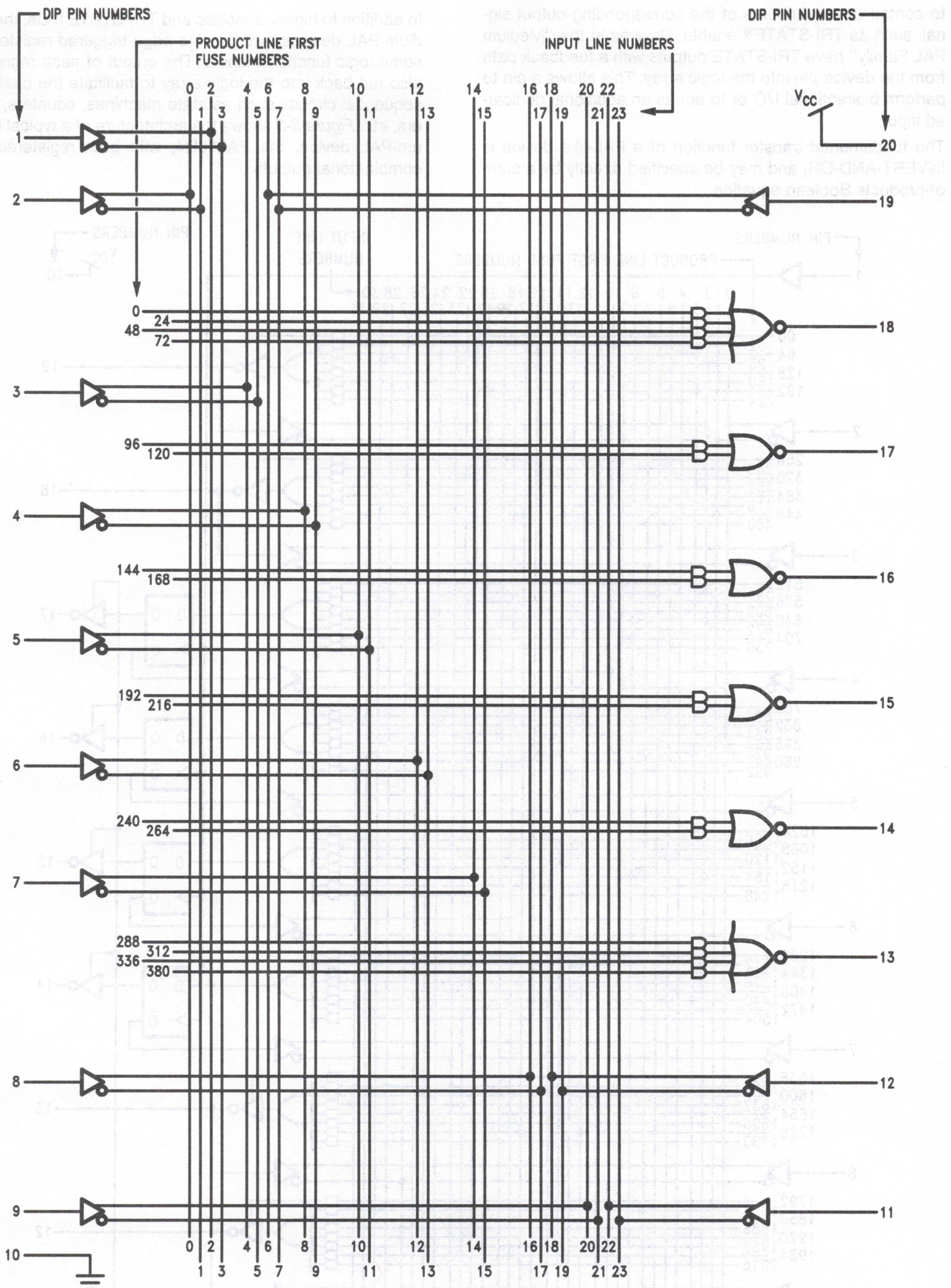
LOGIC ARRAY STRUCTURE

The PAL architecture is based on a single programmable AND-gate array with fixed OR-gate connections. The AND array consists of a number of "input lines" running in one dimension across a number of "product-term lines" running in the orthogonal dimension with programmable interconnection cells at all intersections. For every input signal (logical input variable) applied to the array, a true and complement pair of input lines are provided.

A product term is satisfied (logically true) while all input lines connected to it (via programmable cells) are high. If neither the true nor complement of an array input is connected to a product line, then that array input represents a "don't care" value with respect to that product term.

In all PAL-based devices covered in this book, all product terms are dedicated to specific device outputs. The number of product terms allocated to each output logic function varies from device to device.

In many devices, all of the product terms allocated to each output are simply ORed together to produce the output logic function. The original PAL devices, which have become known as the "Small PAL Family", are based on this simple architecture alone. *Figure 3-2* shows a representative Small-PAL architecture, that of the PAL12H6. Such devices are commonly used for address decoding, multiplexers and random control logic applications.


DIP PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
DIP PIN NUMBERS
V_{CC}
0 2 4 6 8 10 12 14 16 18 20 22
1 3 5 7 9 11 13 15 17 19 21 23
1
2
19
20
0
24
48
72
18
3
96
120
17
4
144
168
16
5
192
216
15
6
240
264
14
7
288
312
336
380
13
8
12
9
11
10

TL/L/9987-4

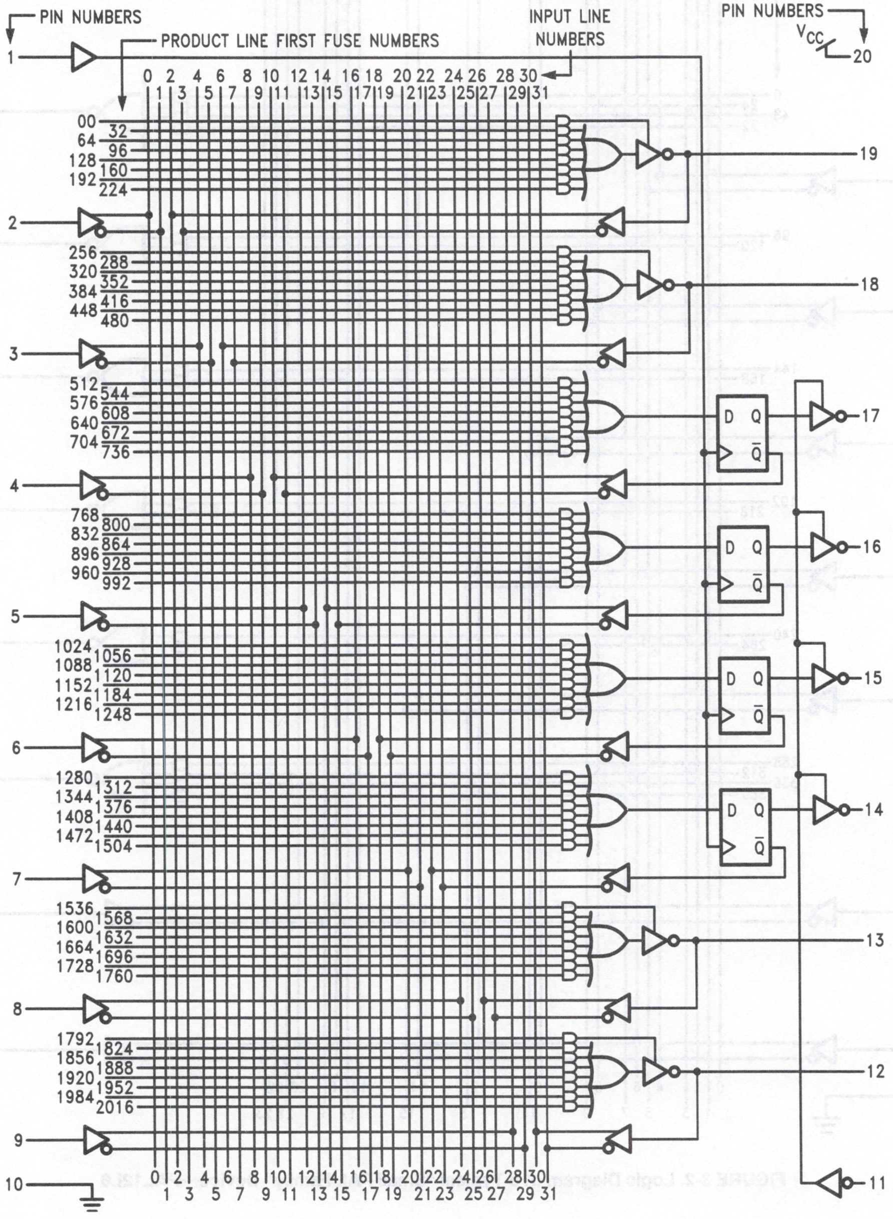
FIGURE 3-2. Logic Diagram of a Typical "Small PAL Family" Device—PAL12L6

In other devices, one or more of the product lines are used to control other attributes of the corresponding output signal, such as TRI-STATE® enable. Devices in the "Medium PAL Family" have TRI-STATE outputs with a feedback path from the device pin into the logic array. This allows a pin to perform bidirectional I/O or to act as an additional dedicated input.

The fundamental transfer function of a PAL-like device is INVERT-AND-OR, and may be specified directly by a sum-of-products Boolean equation.

REGISTERED OUTPUTS

In addition to higher densities and TRI-STATE I/Os, the Medium-PAL devices also provide edge-triggered registers on some logic function outputs. The output of each register is also fed back into the logic array to facilitate the design of sequential circuits such as state machines, counters, shifters, etc. *Figure 3-3* shows the architecture of a typical Medium-PAL device, the PAL16R4, with both registered and combinatorial outputs.

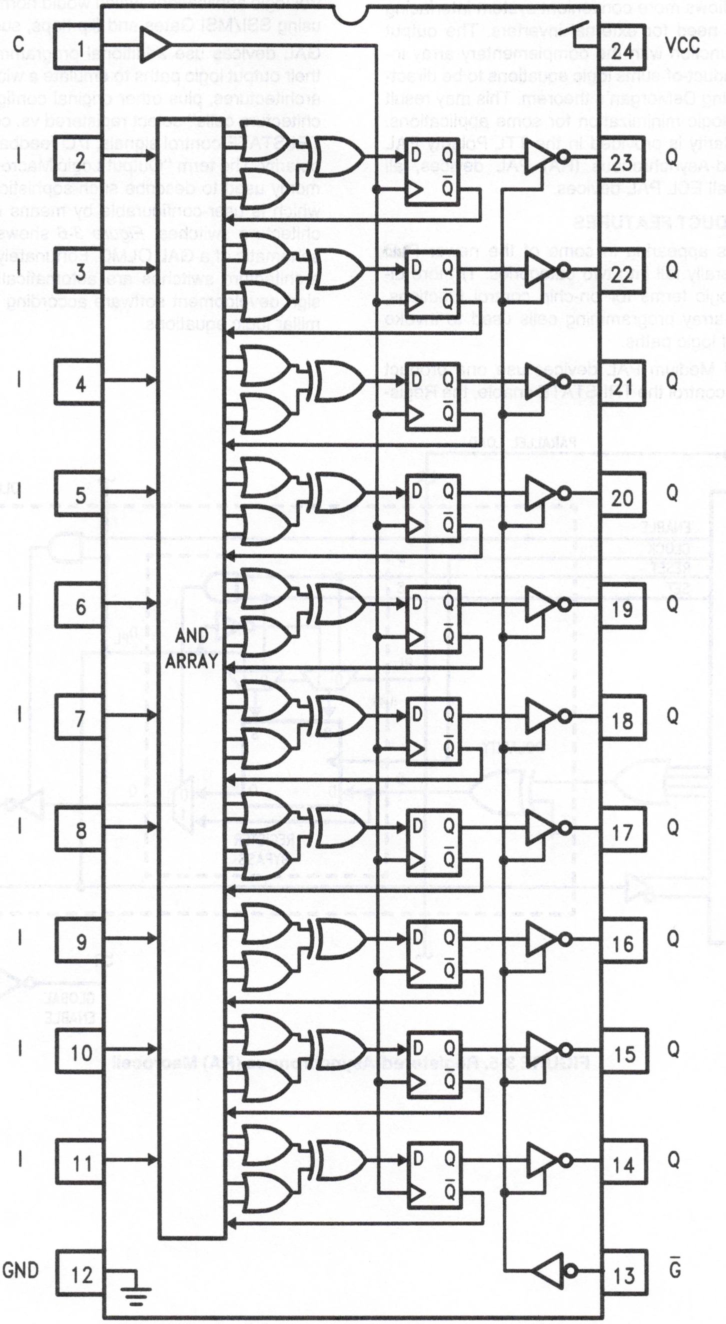

PIN NUMBERS
PRODUCT LINE FIRST FUSE NUMBERS
INPUT LINE NUMBERS
PIN NUMBERS
VCC
20
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
00 32 64 96 128 160 192 224
256 288 320 352 384 416 448 480
512 544 576 608 640 672 704 736
768 800 832 864 896 928 960 992
1024 1056 1088 1120 1152 1184 1216 1248
1280 1312 1344 1376 1408 1440 1472 1504
1536 1568 1600 1632 1664 1696 1728 1760
1792 1824 1856 1888 1920 1952 1984 2016
1
2
3
4
5
6
7
8
9
10
19
18
17
16
15
14
13
12
11
D Q
Q̄

TL/L/9987-7

FIGURE 3-3. Typical "Medium PAL Family" Device—PAL16R4

EXCLUSIVE-OR GATES

Some PAL architectures add an XOR gate, combining two small AND-OR functions, to each output, as shown in *Figure 3-4*. The XOR gates combined with output registers facilitate design of counters, state machines and small arithmetic functions.


C
1
VCC
24
I
2
Q
23
I
3
Q
22
I
4
Q
21
I
5
Q
20
I
6
Q
19
AND
ARRAY
I
7
Q
18
I
8
Q
17
I
9
Q
16
I
10
Q
15
I
11
Q
14
GND
12
13
G̅
D
Q
Q̅

TL/L/9987-8

FIGURE 3-4. PAL20X10 Block Diagram

PROGRAMMABLE OUTPUT POLARITY

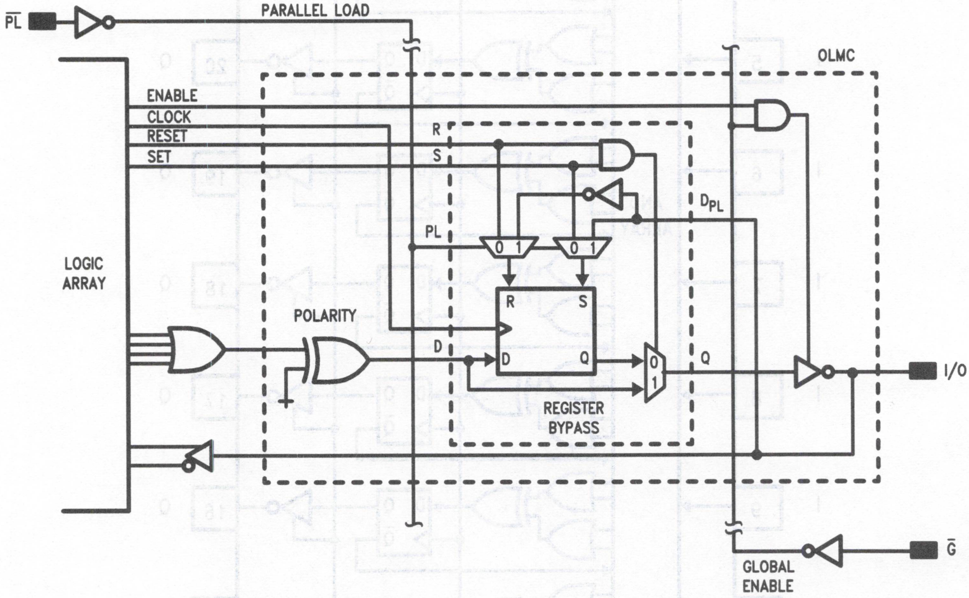
Several more recent PLD families provide programmable output polarity. Additional programming cells are included in the output logic paths to optionally invert the individual output signals. This allows more convenient system interfacing by eliminating the need for external inverters. The output inversions in conjunction with the complementary array inputs also allow product-of-sums logic equations to be directly implemented using DeMorgan's theorem. This may result in more compact logic minimization for some applications. Programmable polarity is provided in the TTL Polarity PAL Family, Registered-Asynchronous (RA) PAL devices, all GAL devices and all ECL PAL devices.

ADVANCED PRODUCT FEATURES

Advanced features appearing in some of the newer PLD architectures generally fall into two categories: 1) more diversified use of logic terms for on-chip control functions, and 2) more non-array programming cells used to invoke optional peripheral logic paths.

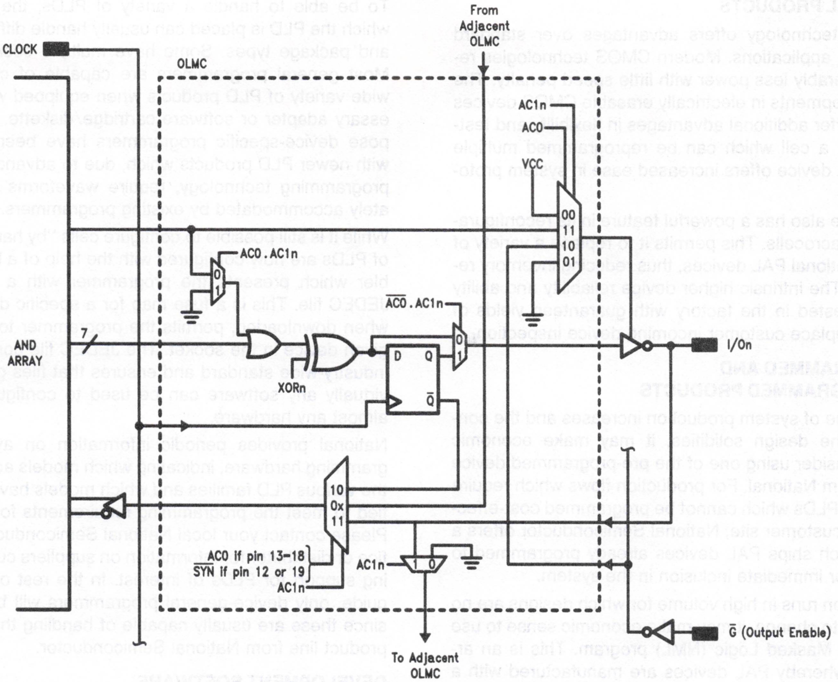
Just as traditional Medium PAL devices use one product term per output to control the TRI-STATE enable, the Registered-Asynchronous devices, PAL16RA8 and PAL20RA10, use three additional "specialized" product terms to control the clocking, reset and preset of each output register *(Figure 3-5)*. This type of PLD is primarily useful in random control logic applications which would normally be implemented using SSI/MSI Gates and flip-flops, such as 74LS74.

GAL devices use additional programming cells to redirect their output logic paths to emulate a wide variety of TTL PAL architectures, plus other original configurations. These "architecture cells" select registered vs. combinatorial outputs, TRI-STATE control signals, I/O feedback paths, and output polarity. The term "Output Logic Macrocell" (OLMC) is commonly used to describe such sophisticated peripheral logic which is user-configurable by means of programmable architecture switches. *Figure 3-6* shows the complete logic schematic of a GAL OLMC. Fortunately, the logic paths and architecture switches are automatically configured by design development software according to the designer's familiar logic equations.



TL/L/9987-5

FIGURE 3-5. Registered-Asynchronous (RA) Macrocell



TL/L/9987-6

FIGURE 3-6. GAL Output Logic Macrocell (OLMC)

3.6 Programmable Logic from National

National Semiconductor offers a broad range of products in the PLD field. From the original PROM devices, the line has expanded to include both the standard lateral fuse-link and the new vertical-fuse PAL devices and now the highly flexible GAL devices.

TTL PROMS

These memory devices can be used as logic devices in instances where the number of inputs is low and the number of product terms required is high. PROM products most frequently used in logic applications belong to the high-speed bipolar family with depths ranging from 32 to 1024 and widths of either 4 or 8 bits. Detailed information on National PROM products is available in a separate databook.

LATERAL FUSE TTL PAL PRODUCTS

These are the workhorses of the PLD product line. They include devices which offer a wide range of function, price and performance. Implemented in advanced bipolar TTL, they use the more traditional titanium-tungsten (TiW) lateral fuse and are available at prices competitive with standard MSI logic. For most applications requiring moderate functional density and medium to high speed, these offer the optimal cost-effective solution.

VERTICAL FUSE TTL PAL PRODUCTS

This innovation, begun at Fairchild Semiconductor, was developed to address the need for the highest performance required in TTL system environments. A vertical fuse structure on the die not only reduces die size, but also decreases propagation time due to more compact circuitry. Vertical fuse technology is used to implement the highest speed PAL devices ("Series-D" and beyond). At the same time, this technology offers increased reliability, testability and programming yield. A more detailed discussion of the technology involved is given in Section 6.1.

ECL PAL PRODUCTS

A large range of the traditional PAL architectures are available in ECL for those system designers taking advantage of this technology. In very high speed applications where ECL is typically used, logic optimization becomes crucial and the delays involved in off-chip wiring become more pronounced as a fraction of the total delay. The SSI/MSI logic families available in ECL are not as robust as those in TTL and tend to consume larger system board areas in implementation. Both problems can be more effectively reduced through the use of ECL PLDs. Consolidating logic into PLDs moves local logic interconnections on-chip eliminating the associated wire delays. Also, by reducing board area consumed by the logic implemented in the PLD, remaining on-board logic becomes more compact thereby reducing the trace lengths of interconnections among surrounding logic. This adds to the speed advantage. Reducing required board area also allows the system designer to increase the amount of on-board logic.

E^{2}CMOS GAL PRODUCTS

The CMOS technology offers advantages over standard TTL in some applications. Modern CMOS technologies require considerably less power with little speed penalty. The recent developments in electrically erasable CMOS devices (E^{2}CMOS) offer additional advantages in flexibility and testability. Using a cell which can be reprogrammed multiple times, a GAL device offers increased ease in system prototyping.

A GAL device also has a powerful feature in its reconfigurable output macrocells. This permits it to replace a variety of more conventional PAL devices, thus reducing inventory requirements. The intrinsic higher device reliability and ability to be fully tested in the factory with guaranteed yields of 100% can replace customer incoming device inspection.

PRE-PROGRAMMED AND MASK-PROGRAMMED PRODUCTS

As the volume of system production increases and the confidence in the design solidifies, it may make economic sense to consider using one of the pre-programmed device programs from National. For production flows which require quantities of PLDs which cannot be programmed cost-effectively at the customer site, National Semiconductor offers a program which ships PAL devices already programmed to customers for immediate inclusion in the system.

For production runs in high volume for which designs are no longer liable to change, it may make economic sense to use the National Masked Logic (NML) program. This is an arrangement whereby PAL devices are manufactured with a given logic configuration already fixed in the metallization step. The relationship of NML to PLDs is similar to that of ROMs to PROMs. Eliminating all dependency on programmable cells (fuses) and the supporting programming circuitry means less on-chip circuitry is liable to cause device failure. The resulting increase in functional yield at the factory allows NML devices to be offered at more competitive prices with respect to standard PLDs, once initial engineering costs are recovered.

Both programs avoid the manufacturing costs associated with device programming and related programming failures. They also considerably enhances device reliability due to the thorough factory test performed on functional devices before shipment using the customer's test patterns. Because of the resulting increase in product quality levels, further manufacturing costs may be saved by eliminating incoming component inspection.

3.7 Design Development Tools

Design development with PLDs involves the use of a few tools which assist both in the design conception and implementation stages. A more detailed discussion of design tools is given in Section 5. Competition in the PLD support market has considerably reduced the investment necessary to acquire the appropriate hardware and software tools.

PROGRAMMING HARDWARE

PLDs are typically configured on a piece of hardware known as a programmer. The selection has widened due to the proliferation of PLD types and now includes an extensive variety of both device-general and device-specific programmers. Several major manufacturers sell general programmers, either as a stand-alone system or as an add-in to an existing computer such as an IBM® PC, and in a broad range of feature complements and price.

To be able to handle a variety of PLDs, the socket into which the PLD is placed can usually handle different device and package types. Some have multiple sockets built in. Most general programmers are capable of configuring a wide variety of PLD products when equipped with the necessary adapter or software cartridge/diskette. Special-purpose device-specific programmers have been introduced with newer PLD products which, due to advanced circuit or programming technology, require waveforms not immediately accommodated by existing programmers.

While it is still possible to configure cells "by hand", the bulk of PLDs are now configured with the help of a logic assembler which presents the programmer with a standard or JEDEC file. This is a fuse map for a specific device which, when downloaded, permits the programmer to configure a given device in the socket. The JEDEC file conforms to an industry-wide standard and ensures that files generated by virtually any software can be used to configure PLDs on almost any hardware.

National provides periodic information on available programming hardware, indicating which models accommodate the various PLD families and which models have been certified to meet the programming requirements for NSC parts. Please contact your local National Semiconductor sales office or distributor for information on suppliers currently offering support for PLDs of interest. In the rest of this design guide, only device-general programmers will be discussed since these are usually capable of handling the entire PLD product line from National Semiconductor.

DEVELOPMENT SOFTWARE

These are software packages available for certain computers which allow the user to enter the design through the keyboard in a number of ways. Most packages permit the use of Boolean logic equations and truth tables. Assemblers are from specific PLD suppliers, programmer vendors or third party vendors.

NATIONAL SEMICONDUCTOR PLAN™ SOFTWARE

National Semiconductor offers a PLD assembler known as PLAN software, which is a package optimized for use with the PLDs available from NSC and includes most industry-standard PAL architectures. PLAN software is available to operate primarily on IBM-compatible PC under MS-DOS and uses Boolean equations as the method of design entry. Unlike others, the PLAN assembler allows designs to be specified independent of specific device architectures and will optionally select an appropriate device which will accommodate the final design. Refer to Section 5.3 for a more in-depth discussion of the PLAN package.

OTHER ASSEMBLERS/COMPILERS

High-level assemblers provide many automatic features that are not found in other assemblers. They are usually device and manufacturer-independent and some tools may be usable with other software packages. While they are still capable of accepting Boolean equation data entry, they offer other input alternatives, such as gate-level graphics and higher-level state machine descriptive language, and allow features such as set-definition and automated logic reduction. Several higher-level packages designed for use with all PLDs are generally available. Two of the most popular are CUPL® software from Logical Devices and ABEL® software from Data I/O.

Section 4

Programmable Logic Design Methodology

Section 4 Contents

4.1 Design Development Process 4-3
4.2 Logic Design 4-3
  Defining the problem
  Designing the logic
  Deriving Boolean equations
  Logic minimization
4.3 Design Implementation 4-6
  Selecting a device
  Design/equation entry
  Compilation—creating the JEDEC file
  Programming hardware preparation
  Device programming
4.4 Logic Verification 4-8
  Pattern verfication
  Test vector generation
  Design simulation
  Device functional testing

National Semiconductor

Section 4
Programmable Logic Design Methodology

PLDs offer a number of design advantages to the designer. But in order to make use of these, the designer must apply a specific design methodology in order to maximize the effectiveness of the PLD tools available. This chapter outlines the steps by which this is done and illustrates their use by specific examples. Details of the use of the most important tools are given in Section 5. For more diverse examples without as much emphasis on methodology, refer to Section 7—Applications.

4.1 Design Development Process

The design development process for PLDs proceeds in three main phases:

- Logic design
- Design implementation
- Design/Logic verification

Within each phase, an experienced designer will pass through a number of steps. While the process may appear involved at first, it is mostly a stylization of good design practices and efficient use of the PLD tools available to the designer.

Within the Logic Design phase, the steps involved are common procedure for any digital design engineer and can be tailored to suit the individual taste and design requirements. The main steps are:

- Define the system problem
- Generate a block diagram
- Implement the function logically
- Derive the Boolean equations describing the design

These are largely self-explanatory. For readers requiring some background information on logic design principles, refer to Appendix A.

Design Implementation consists largely of selecting and using the tools to translate the results of the first phase into a configured PLD. It consists of steps:

- PLD family and device selection
- Partitioning the logic to fit the devices selected
- Equation entry
- Running development software and JEDEC file creation
- Platform and programmer configuration
- JEDEC file transfer
- Device programming

The preparation of software, platform and programmer need be done ony for the initial use of PLDs. Following that, the other functions are all straightforward operations often handled automatically by the PLD tools selected and not requiring involvement on the part of the designer.

Design verification is the final phase during which the correct programming of the device is checked, along with the generation of test procedures which verify that the device itself implements what was originally required. The steps in this phase are:

- Device programming verification
- Design test vector generation
- Device simulation
- Device functional test
- Design documentation

The effort involved in each depends both on the design complexity and the tools available. As with any other design, the verification phase can be too easily overlooked in the entire design process, but effort spent in judicious testing and adequate documentation is normally well spent.

Each of these steps is described in detail later in this section.

4.2 Logic Design

This section gives a detailed account of the steps involved in generating the initial theoretical design, illustrated by reference to an example of a 6-bit bidirectional shift register.

DEFINING THE PROBLEM

As with any other design methodology, the first step involved is a clear definition of the problem to be solved. In the case of the shift device, what is required is a device with the following characteristics:

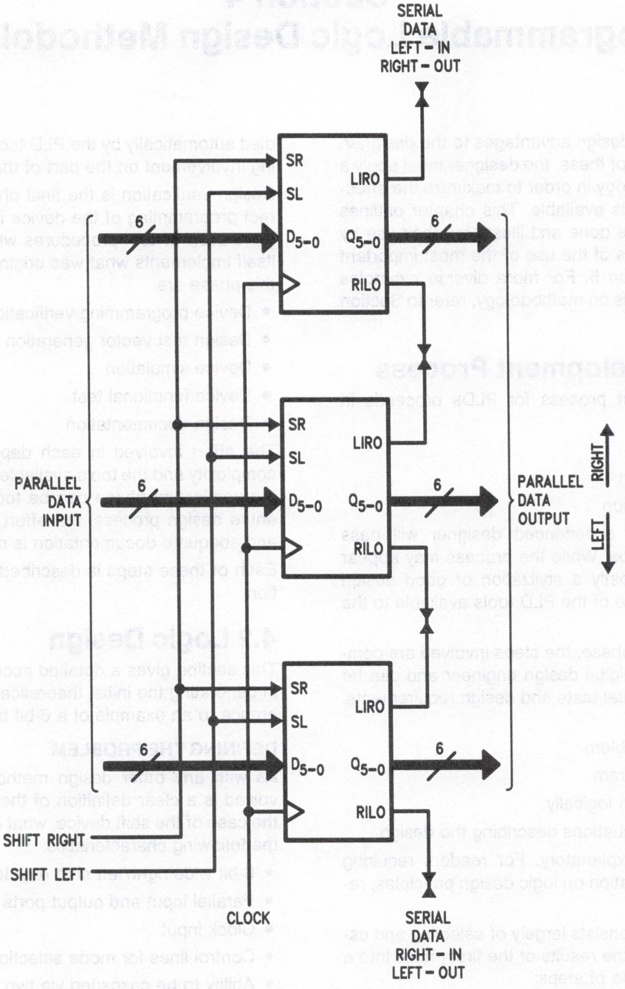
- 6-bit wide right/left shift register
- Parallel input and output ports
- Clock input
- Control lines for mode selection
- Ability to be cascaded via two bidirectional serial ports

Additional criteria which might play a role in selection of the final solution are the need for low parts count, power and speed considerations, and the need to interface with or mop up other logic in the area. For the purpose of this example, assume these criteria impose no special constraints.

DESIGNING THE LOGIC

Based on the above criteria, the block diagram of the logic can be generated directly, as shown in *Figure 4-1*. The signal names are given to permit unambiguous reference to their function and any considerations of logic context within the system should be incorporated here.

From the block diagram, the designer derives the detailed functional description of the intended behavioral concept. This may take a number of forms, depending on the application and the preference of the designer. One common method of expressing the detailed operation of a registered application such as this is a function table, which is shown in *Figure 4-2* for reference. Another common method is the use of timing waveforms which are omitted for this example due to its simplicity. The target function is further defined by deriving a detailed logic schematic (as shown in *Figure 4-3*), combinatorial truth table, or direct expression in Boolean equations.


SERIAL DATA LEFT-IN RIGHT-OUT
SR
SL
6
D_{5-0}
LIRO
Q_{5-0}
RILO
PARALLEL DATA INPUT
PARALLEL DATA OUTPUT
RIGHT
LEFT
SHIFT RIGHT
SHIFT LEFT
CLOCK
SERIAL DATA RIGHT-IN LEFT-OUT

TL/L/9988-1

FIGURE 4-1. Block Diagram Showing 3 Cascaded Shift Registers

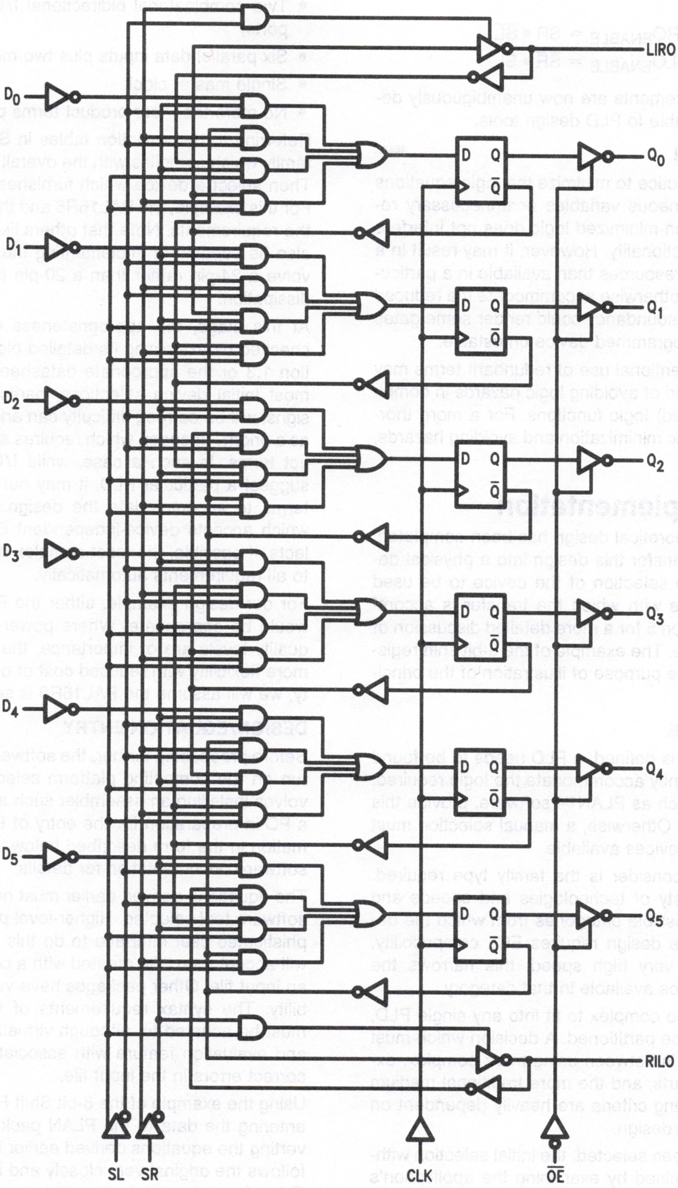
Function	SL	SR	RILO	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	LIRO
Hold	0	0	Z	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	Z
Shift Right	0	1	RI	RI	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
Shift Left	1	0	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	LI	LI
Parallel Load	1	1	Z	D_5	D_4	D_3	D_2	D_1	D_0	Z

FIGURE 4-2. Functional Operation Table for Shifter Example

DERIVING BOOLEAN EQUATIONS

In order to provide a definition of the circuit which the design tools can handle, it is usually necessary to express the design in terms of Boolean equations. The fundamental transfer function of a PAL® device is the sum-of-products or, through DeMorgan inversion, product-of-sums form. Logic equations can be derived directly from the function table shown in *Figure 4-2*, the logic schematic in *Figure 4-3*, or from the method of logic implementation preferred.

For any TRI-STATE output, including bidirectional I/O lines, additional equations may need to be specified to define the control functions of these lines. This is illustrated in the example of the 6-bit shift register.



TL/L/9988-2

FIGURE 4-3. Gate-Level Logic Schematic of Shifter

For the 6-bit shift register example, the equations are:

$$Q0 \leftarrow Q_0 \bullet \overline{SR} \bullet \overline{SL} + Q_1 \bullet SR \bullet \overline{SL} + LIRO \bullet \overline{SR} \bullet SL$$
$$+ D_0 \bullet SR \bullet SL$$

$$Q_i \leftarrow Q_i \bullet \overline{SR} \bullet \overline{SL} + Q_{i+1} \bullet SR \bullet \overline{SL} + Q_{i-1} \bullet \overline{SR} \bullet SL$$
$$+ D_i \bullet SR \bullet SL, i = 1 \ldots 4$$

$$Q_5 \leftarrow Q_5 \bullet \overline{SR} \bullet \overline{SL} + RILO \bullet SR \bullet \overline{SL} + Q_4 \bullet \overline{SR} \bullet SL$$
$$+ D_5 \bullet SR \bullet SL$$

$$LIRO_{OUTPUT} = Q_0;\ LIRO_{ENABLE} = SR \bullet \overline{SL}$$
$$RILO_{OUTPUT} = Q_5;\ RILO_{ENABLE} = \overline{SR} \bullet SL$$

The device logic requirements are now unambiguously defined in a form acceptable to PLD design tools.

LOGIC MINIMIZATION

It is generally good practice to minimize the logic equations to eliminate any extraneous variables or unnecessary redundant min-terms. Non-minimized logic does not interfere with proper device functionality. However, it may result in a design requiring more resources than available in a particular device which could otherwise accommodate the reduced equations. Also, logic redundancy could render some gates or nodes within the programmed device untestable.

On the other hand, intentional use of redundant terms may be a convenient method of avoiding logic hazards in combinatorial (unsynchronized) logic functions. For a more thorough discussion of logic minimization and avoiding hazards, refer to Appendix A.

4.3 Design Implementation

Now that the basic theoretical design has been completed, the next stage is to transfer this design into a physical device. This requires the selection of the device to be used and a number of tools with which the transfer is accomplished. Refer to Section 5 for a more detailed discussion of PLD tools and their use. The example of the 6-bit shift register is again used for the purpose of illustration of the principles.

SELECTING A DEVICE

Once the logic design is defined, a PLD needs to be found which can most efficiently accommodate the logic required. Some design tools, such as PLAN™ software, provide this function automatically. Otherwise, a manual selection must be made among the devices available.

The first criterion to consider is the family type required. PLDs come in a variety of technologies and speeds and offer a spectrum of possible categories from which the device is selected. If the design requires ECL compatibility, CMOS low-power or very high speed, this narrows the choice down to a device available in that category.

If the target logic is too complex to fit into any single PLD, then the design must be partitioned. A decision which must be made at this point is between using more complex, expensive and slower parts, and the more traditional medium PAL devices. Partitioning criteria are heavily dependent on the goals of a specific design.

Once the family has been selected, the initial selection within the family is determined by examining the application's block diagram, function table and logic equations. Based on these, the following parameters are established:

- Number of registered outputs required
- Number of combinatorial outputs required
- Number of inputs required
- Clocking requirements
- Complexity of each logic equation (number of min-terms required)

For our 6-bit shift register, it can be seen that the requirements are:

- Six registered outputs (for the parallel-out lines)
- Two combinatorial bidirectional I/O lines (for the serial ports)
- Six parallel data inputs plus two mode control inputs
- Single master clock
- No more than four product terms per output function

Referring to the selection tables in Section 1.1, select the family which complies with the overall system requirements. Then select a device which furnishes all the requirements. For this example, the PAL16R6 and the GAL16V8 both fulfill the requirements. Note that others like the PAL20R6 would also be capable of implementing the design, but would involve a 24-pin rather than a 20-pin part and higher power dissipation.

At this stage, the appropriateness of the device is also checked by examining its detailed block diagram (see Section 1.3 or the appropriate datasheet in Section 2). While most initial device selections, particularly for simpler designs, will be correct, difficulty can arise with a design, such as a priority encoder, which requires a large number of product terms. In such a case, while I/O requirements might suggest a particular PLD, it may not offer enough product terms to accommodate the design. The PLAN software, which accepts device-independent Boolean equations selects, if possible, the least complex device which conforms to all requirements automatically.

For our design example, either the PAL16R6 or GAL16V8 would be appropriate. Where power-dissipation or highest quality levels are of importance, the GAL® solution offers more flexibility with reduced cost of ownership. For simplicity, we will assume the PAL16R6 is selected.

DESIGN/EQUATION ENTRY

Before proceeding further, the software tools will need to be run on the computing platform selected. This normally involves installing an assembler such as PLAN software onto a PC in preparation for the entry of the programming information in the form described below. Refer to the individual software documentation for details.

The equations derived earlier must now be entered into the software tool selected. Higher-level packages provide a sophisticated user interface to do this. The PLAN assembler will accept a text file created with a common editor utility as an input file. Other packages have varying degrees of flexibility. The syntax requirements of the software package must be adhered to, although virtually all provide a parsing and evaluation feature with associated error messages to correct errors in the input file.

Using the example of the 6-bit Shift Register, the method of entering the data to the PLAN package involves first converting the equations derived earlier into PLAN syntax. This follows the original very closely and is shown in *Figure 4-4*. This information can be generated using any convenient editor, such as EDLIN, or even a word processing package.

```
PAL16R6; 6-BIT CASCADABLE SHIFT REGISTER

;Pin List:
CLK SR   D0 D1 D2 D3 D4 D5 SL   GND
/G  RILO Q5 Q4 Q3 Q2 Q1 Q0 LIRO VCC

;Equations:
/Q0  := /Q0*/SR*/SL +   /Q1*SR*/SL + /LIRO*/SR*SL + /D0*SR*SL
/Q1  := /Q1*/SR*/SL +   /Q2*SR*/SL +    /Q0*/SR*SL + /D1*SR*SL
/Q2  := /Q2*/SR*/SL +   /Q3*SR*/SL +    /Q1*/SR*SL + /D2*SR*SL
/Q3  := /Q3*/SR*/SL +    /Q4*SR*/SL +   /Q2*/SR*SL + /D3*SR*SL
/Q4  := /Q4*/SR*/SL +    /Q5*SR*/SL +   /Q3*/SR*SL + /D4*SR*SL
/Q5  := /Q5*/SR*/SL + /RILO*SR*/SL +   /Q4*/SR*SL + /D5*SR*SL
/LIRO = /Q0
LIRO.TRST = SR*/SL
/RILO = /Q5
RILO.TRST = /SR*SL
```

TL/L/9988–4

FIGURE 4-4. PLAN™ Input File for Shifter Example

```
PLAN v3.12   09-02-1988    14:19
Source filename: SHIFTER.BEQ    Device: PAL16R6
6-BIT CASCADABLE SHIFT REGISTER *
QP20* QF2048* F0*
L0000
01111111111111111111111111111011
11111101111111111111111111111111*
L0256
10111110111111111111111111111011
01111111111011111111111111111011
10101111111111111111111111110111
01111011111111111111111111110111*
L0512
10111111111011111111111111111011
01111111111111101111111111111011
10111110111111111111111111110111
01111111101111111111111111110111*
L0768
10111111111111101111111111111011
01111111111111111110111111111011
10111111111011111111111111110111
01111111111110111111111111110111*
L1024
10111111111111111110111111111011
01111111111111111111111011111011
10111111111111101111111111110111
01111111111111111011111111110111*
L1280
10111111111111111111111011111011
01111111111111111111111111101011
10111111111111111110111111110111
01111111111111111111101111110111*
L1536
10111111111111111111111111101011
01111111111111111111111111111010
10111111111111111111111011110111
01111111111111111111111110110111*
L1792
10111111111111111111111111110111
11111111111111111111111111101111*
C6743* DC06
```

TL/L/9988–5

FIGURE 4-5. JEDEC File for Shifter Example Produced by PLAN Software

COMPILATION—CREATING THE JEDEC FILE

At this stage, pin assignment is normally made, either manually or automatically by the software. PLAN offers an automatic assignment, which can then be edited manually, if desired.

Once the equation file has been entered and pin assignments resolved, the assembly is performed on the platform, the results of which are a JEDEC fuse map for down-loading to the programmer.

In order for the equations to be converted into a bit pattern from which the cells of a PLD can be systematically programmed, the initial equations must be converted into a form, known as the JEDEC file. The JEDEC file is an industry-wide standard accepted by all programming hardware. It consists of a formatted table indicating all of the cells (fuses) in the PLD to be programmed to implement the specified logic functions. This is done by a module within the software tool known as the assembler. Details of operation of the assembler varies from one package to another, but each provides syntax checking and an evaluation of whether the design can be implemented in the device chosen, as well as the JEDEC file itself, which is normally stored on disk ready for down-loading to a programmer.

The actual form of the JEDEC file is usually of little interest to the system designer. Its only purpose is to provide a uniform interface between commercial PLD software and hardware tools and no real information for the designer is provided by its details. It may occasionally be useful as a debugging aid to isolate any problems occurring between equation entry and functional test.

PROGRAMMING HARDWARE PREPARATION

Before the cell data can be transferred, the programmer needs to be connected to the software platform and fitted with any socket adapters required to accommodate the blank sample device.

For the purpose of this example, a PAL16R6 is being programmed with a Data I/O Model 29 programmer equipped with Logic-Pack. In order to do this, connect the System 29 to the platform via an RS232C cable, according to the system documentation. An outline of this is given in Section 5. A PAL16R6 requires the 303A-011A adapter to be installed on the Logic-Pack. Look up the PAL16R6 on the device chart to determine and enter the family and pinout code.

DEVICE PROGRAMMING

This step involves transferring the prepared JEDEC file across a communication link from the platform to the hardware programmer. Each programmer differs slightly, but generally each requires that a number of prompts be answered with such information as file name, device type and manufacturer. Usually, a test is run at this time on the device by the programmer which ensures that the device is correctly oriented in the socket and is in fact blank and able to be programmed. The correctness of data transfer is verified by means of a checksum transmitted with the file.

For the purposes of our example, the System 29 provides all the prompts required to do this.

Now that all relevant information has been entered into the programmer, it is a matter of simply invoking the Program function.

This translates the JEDEC file into addresses, data patterns and programming pulses, which, when applied to the pins of the device in the socket, will configure the cells of the device in a pattern which will cause the device to operate in accordance with the original design. The implementation of the design in the PLD has now been completed.

Again, for the purposes of our example, the System 29 provides all the prompts required to do this.

4.4 Logic Verification

Verification is required to ensure not only that the device has been configured exactly as intended, but also that the programmer has functioned correctly and that the design performs as originally intended. Again, this takes the form of several steps.

PATTERN VERIFICATION

This may be performed automatically by the programmer. If not, it is recommended that a manual verification run is performed while the device is still in the programmer to ensure that the pattern set in the device corresponds to that specified by the fuse map in the JEDEC file. This is a simple step which is done by the programmer itself. The programmer reads the pattern directly from the PLD, similar to reading a PROM, and compares this directly with the original JEDEC file still resident in the programmer.

TEST VECTOR GENERATION

Particularly with more complex designs, it is recommended that some consideration for a set of device exercises, generally known as test vectors, be given as early as the equation entry stage above. Some advanced software tools may provide automatic generation of test vectors from the equations as they are entered. Otherwise, vectors must be generated by hand. Even in the case of automatic test vector generation, some designers prefer to add their own additional vectors to verify application-specific operations.

For a design of the complexity of our 6-bit shift register, test vectors are easily generated by hand, as shown in *Figure 4-6*. In this case, a test such as walking ones and zeroes is probably sufficient to prove functionality of the device beyond reasonable doubt. For more complex designs, it may be helpful to employ fault-grading software to ensure adequate coverage of all design paths and gates by the test vectors.

DESIGN SIMULATION

This optional step generates the device output vectors which allow verification of correct design operation. This can be done manually, or with the help of the simulator module of the software tool, to predict the output configuration for the device based on the original software model entered as Boolean equations.

The output vectors from the simulation must be examined to confirm that the model operates correctly. They also provide the output states, which must accompany the test vectors for functional testing of the device.

Beyond device-level simulation, additional software is becoming available to generate models of the programmed PLDs for use in system-level simulations. Such simulations are typically performed on CAD workstations and, more recently, personal computers.

DEVICE FUNCTIONAL TESTING

The device is evaluated fully for correct performance of the function desired. In the case of the 6-bit shift register, this would involve checking all the functions outlined in the function table in *Figure 4-2*.

Varying from one software tool to another, the test vectors are entered (if not generated automatically) into the software, which appends them in the proper format to the JEDEC file, as shown in *Figure 4-7*. The test vector entry/generation typically follows the equation entry step, so that the combined JEDEC file is down-loaded to the programmer. Later, following device programming and pattern verification steps, the programmer performs the functional test on the PLD while still in the socket. The input vector waveforms are applied to the device pins while in the normal operational mode, and the device output signals are compared with the expected output vectors.

As a final step, most PLDs include a "security cell/fuse" which, when programmed, disables further programming and verifying. This prevents direct copying of the logic patterns resulting in proprietary custom circuits that are difficult to copy or reverse engineer.

Inputs								Bidirectional I/O		Outputs						Comments
SL	SR	D_5	D_4	D_3	D_2	D_1	D_0	LIR0	RIL0	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0	
1	1	0	0	0	0	0	0	1	1	L	L	L	L	L	L	Load all zeroes.
0	0	1	1	1	1	1	1	1	1	L	L	L	L	L	L	Hold zeroes.
1	0	1	1	1	1	1	1	1	L	L	L	L	L	L	H	Shift left single one,
1	0	1	1	1	1	1	1	0	L	L	L	L	L	H	L	followed by zeroes.
1	0	1	1	1	1	1	1	0	L	L	L	L	H	L	L	
1	0	1	1	1	1	1	1	0	L	L	L	H	L	L	L	
1	0	1	1	1	1	1	1	0	L	L	H	L	L	L	L	One shifts out of RIL0,
1	0	1	1	1	1	1	1	0	H	H	L	L	L	L	L	and vanishes.
1	0	1	1	1	1	1	1	0	L	L	L	L	L	L	L	Load all ones.
1	1	1	1	1	1	1	1	0	0	H	H	H	H	H	H	Hold ones.
0	0	0	0	0	0	0	0	0	0	H	H	H	H	H	H	Shift right single zero,
0	1	0	0	0	0	0	0	H	0	L	H	H	H	H	H	followed by ones.
0	1	0	0	0	0	0	0	H	1	H	L	H	H	H	H	
0	1	0	0	0	0	0	0	H	1	H	H	L	H	H	H	
0	1	0	0	0	0	0	0	H	1	H	H	H	L	H	H	
0	1	0	0	0	0	0	0	H	1	H	H	H	H	L	H	
0	1	0	0	0	0	0	0	L	1	H	H	H	H	H	L	Zero shifts out of LIR0,
0	1	0	0	0	0	0	0	H	1	H	H	H	H	H	H	and vanishes.

Key: 0 = Apply Low Input, 1 = Apply High Input
L = Expect Low Output, H = Expect High Output

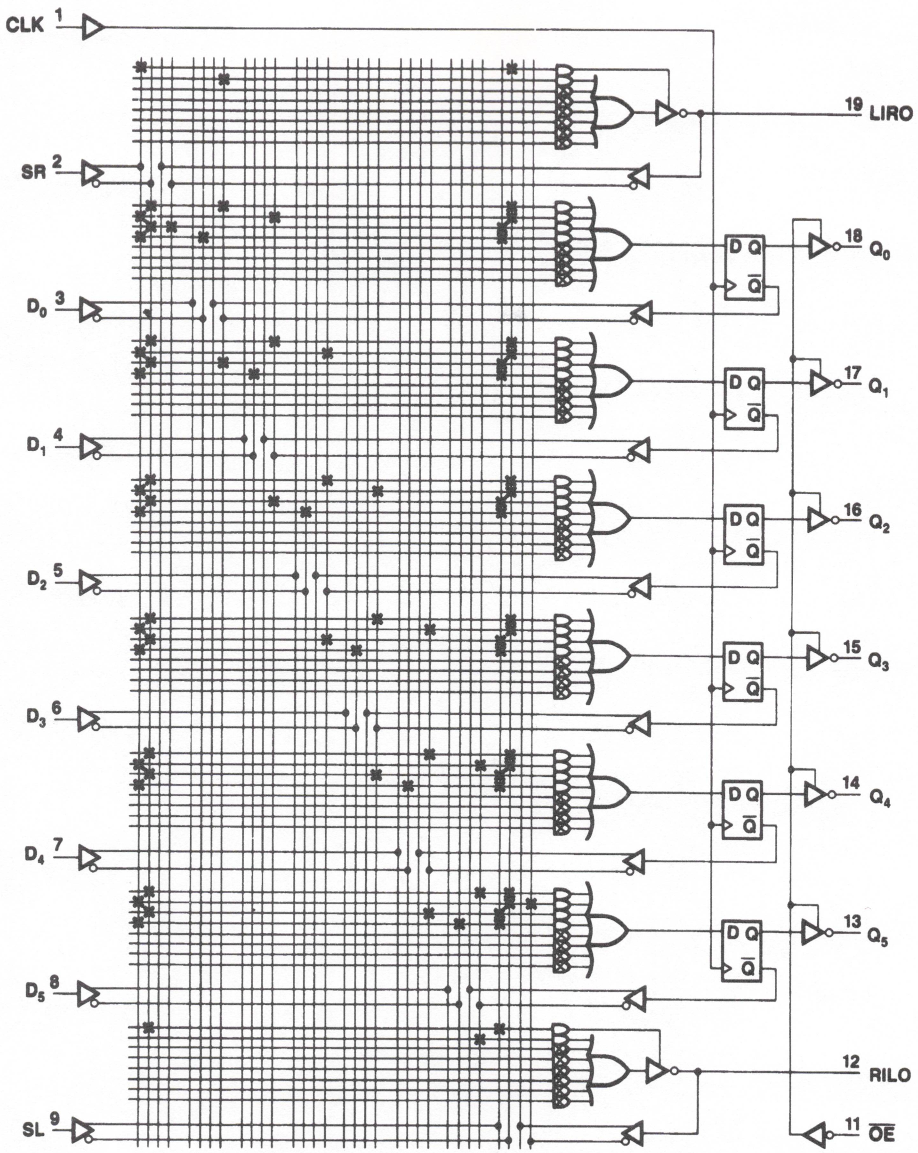
Note: The device is clocked after applyling each input vector. Outputs on the same line are strobed and compared after the clock.

FIGURE 4-6. Functional Test Pattern for 6-Bit Shift Register Example

```
PLAN v3.12   09-02-1988    14:19
Source filename: SHIFTER.BEQ     Device: PAL16R6
6-BIT CASCADABLE SHIFT REGISTER *
QP20* QF2048* F0*
L0000
01111111111111111111111111111011
11111110111111111111111111111111*
L0256
10111110111111111111111111111011
01111111111011111111111111111011
10101111111111111111111111110111
01111011111111111111111111110111*
L0512
10111111111011111111111111111011
01111111111111101111111111111011
10111110111111111111111111110111
01111111101111111111111111110111*
L0768
10111111111111101111111111111011
01111111111111111110111111111011
10111111111011111111111111110111
01111111111110111111111111110111*
L1024
10111111111111111110111111111011
01111111111111111111111011111011
10111111111111101111111111110111
01111111111111111011111111110111*
L1280
10111111111111111111111011111011
01111111111111111111111111101011
10111111111111111110111111110111
01111111111111111111101111110111*
L1536
10111111111111111111111111101011
01111111111111111111111111111010
10111111111111111111111011110111
01111111111111111111111110110111*
L1792
10111111111111111111111111110111
11111111111111111111111111101111*
V0001 0XXXXXXXXN0XXXXXXXXN*
V0002 C10000001N01LLLLLL1N*
V0003 C01111110N01LLLLLL1N*
V0004 C01111111N0LLLLLLH1N*
V0005 C01111111N0LLLLLHL0N*
V0006 C01111111N0LLLLHLL0N*
V0007 C01111111N0LLLHLLL0N*
V0008 C01111111N0LLHLLLL0N*
V0009 C01111111N0HHLLLLL0N*
V0010 C01111111N0LLLLLLL0N*
V0011 C11111111N00HHHHHH0N*
V0012 C00000000N00HHHHHH0N*
V0013 C10000000N00LHHHHHHN*
V0014 C10000000N01HLHHHHHN*
V0015 C10000000N01HHLHHHHN*
V0016 C10000000N01HHHLHHHN*
V0017 C10000000N01HHHHLHHN*
V0018 C10000000N01HHHHHLLN*
V0019 C10000000N01HHHHHHHN*
C6743* 5480
```

TL/L/9988-6

FIGURE 4-7. JEDEC File Combining Logic Array and Test Vectors for 6-Bit Shifter Example


CLK 1
SR 2
D0 3
D1 4
D2 5
D3 6
D4 7
D5 8
SL 9
19 LIRO
18 Q0
17 Q1
16 Q2
15 Q3
14 Q4
13 Q5
12 RILO
11 OE
D Q
Q

TL/L/9988-3

FIGURE 4-8. PAL16R6 Logic Diagram Showing Fuse Pattern of Shifter Example

Section 5

PLD Design Development Tools

Section 5 Contents

5.1 Selecting Development Tools 5-3
  Programming hardware selection
  Selecting the development software
5.2 PLD Programmers 5-3
  Types of programmers
  Programmer certification
  Data I/O Model 29
  Data I/O Unisite
  Logical Devices ALLPRO
  Logic Devices PALPRO-2X
  Stag Microsystems PPZ
  Stag Microsystems ZL30A
5.3 National PLAN Software 5-5
  Features of PLAN software
  PLAN software design entry
  Design compilation
  Back translation
  PAL-to-GAL map conversion
5.4 Other Assemblers/Compilers 5-8
  PALASM software
  ABEL software
  CUPL software
5.5 Pre-Assembler Software Tools 5-8

National Semiconductor

Section 5
PLD Design Development Tools

5.1 Selecting Development Tools

PROGRAMMING HARDWARE SELECTION

The hardware tool used for PLDs is the programmer which physically configures PLDs. There is a wide selection, but due to the continual change in tool availability, details cannot be kept current in a design guide such as this. However, a listing of several programmer manufacturers and models currently available and in development appears in Table 5-1. For information on current tools available, contact your local National Semiconductor sales office.

Generally, programmers are stand-alone devices which interconnect with the computing platform by means of an RS232C link down which the cell configuration is loaded in the standard form of a JEDEC file. While most programmers handle the standard PLDs, as with software, the more specialized or novel the device chosen, the more restricted is the choice of tools available. Device-specific programmers are typically provided for immediate support of newer PLDs requiring programming algorithms significantly different from those already implemented by general-purpose programmers.

SELECTING THE DEVELOPMENT SOFTWARE

When PLDs were first introduced, fuse maps were derived manually. Now, the convenience of assembly-type development software renders the manual approach obsolete. Before PLD design development begins, the appropriate software tool is selected. The software provides the facility of converting the design into a format understandable by device programmers. Depending on the package chosen, a number of other features are offered as well. Software tools reside in a computer, frequently a PC, known as a platform.

Most common packages will handle all of the standard PLDs and several are flexible enough to handle many of the less common types as well. Generally speaking, the more specialized or novel the device selected, the narrower the range of software tools available. In the case of our example, almost any software will handle the PAL16R6 device.

The PLAN™ software package supports all PLD devices available from National Semiconductor. Ordinarily, newer PLDs with advanced architecture features or original combinations of features take some time to be incorporated into new revisions of general-purpose, third-party software tools. However, software tools provided directly by the PLD manufacturers (such as NSC PLAN software) generally provides support for new products at the time of introduction.

5.2 PLD Programmers

A large selection of programmers are available for use as hardware tools in PLD design and development. They vary in complexity and capability.

TYPES OF PROGRAMMERS

Generally, programmers fall into two categories—device-general and device-specific. Device-general programmers are available from established third-party suppliers, such as Data I/O, and are capable of handling standard PLDs and many specialized ones. This section is concerned mainly with device-general tools as they provide the support necessary for most NSC parts. Device-specific programmers are normally produced by the manufacturer of a non-standard PLD to ensure the availability of tools and are not discussed here in detail. Virtually all programmers conform to the JEDEC fuse map standard.

Device-general programmers have developed from tools which were initially used to configure PROMs. Many of the first-generation programmers show this heritage and require considerable operator intervention to complete the programming function. Developments have since produced tools of increasing sophistication and culminating in systems such as the Data I/O Unisite.

Almost all programmers consist of a stand-alone cabinet, which houses the control electronics and which is connected to the software platform by a communications link for fuse map down-loading in JEDEC file form. The control panel usually includes a keyboard for manual data entry and a univeral connector into which are placed adapters which may contain circuitry, algorithm software and/or sockets for specific device types. More recent programmers now have universal sockets equipped with software-configurable pin drivers, which can handle all PLD package types.

Virtually all programmers require a communications port connection to the platform and the installation of communications software in the platform to provide intelligent control of fuse map file transfer. Some involve circuit cards which are installed into PC-type platforms with direct connection to external programming modules. The details vary with the type of programmer.

PROGRAMMER CERTIFICATION

National Semiconductor maintains a program of programmer certification for use with NSC parts. Tools are classified according to the level of support provided for each family of PLDs from National.

For information on current tools available, please contact your local National Semiconductor sales office for information on the latest listings.

Descriptions of some of the more popular programmers which are able to program most or all of National's PLD product line follow.

Table 5-1. Programmer Manufacturers and Models Available or In Development

Manufacturer	Model	Lateral-Fuse TTL	Vertical-Fuse TTL	E^{2}CMOS GAL	ECL
ADVIN 1050-L E DUANE AV. SUNNYVALE, CA 94086 (408) 984-8600	SAILOR-PAL	●		●	●
DATA I/O CORP BOX 97046 REDMOND, WA 98073 (206) 881-6444, TLX 152167	29B	●	●	●	
	UNISITE 40	●	●	●	●
	60A	●		●	
DIGILEC INC 22736 VANOWEN ST. CANOGA PARK, CA 91307 (818) 887-3755	803-LDC1,2,3	●	●	●	●
	860	●	●	●	●
DIGITAL MEDIA 11770 WARNER AV., STE 225 FOUNTAIN VALLEY, CA 92708 (714) 751-1373	IQ-180	●	●	●	●
	IQ-280	●	●	●	●
INLAB INC 2150-I W 6th AV. BROOMFIELD, CO 80020 (303) 460-0103, TLX 797159	28L	●	●	●	●
LOGICAL DEVICES INC 1201 NW 65th PLACE FORT LAUDERDALE, FL 33309 (305) 974-0967, TLX 383142	ALL PRO	●	●	●	●
	PALPRO-2X	●	●	●	●
MICROPROSS, FRANCE PARC D'ACTIVITE DES PRES 5, RUE DENIS-PAPIN 59650 VILLENUEVE-D'ASCQ TELEX MUPROSS 120611 F		●			●
PROGRAMMABLE LOGIC TECHNOLOGIES INC BOX 1567 LONGMONT, CO 80501 (303) 772-9059	LOGIC LAB			●	
QWERTY INC 5346 BRAGG ST SAN DIEGO, CA 92122 (619) 455-0500	GPR-1000			●	
STAG MICROSYSTEMS INC 1600 WYATT DR SANTA CLARA, CA 95054 (408) 988-1118	ZL30A	●	●	●	
	PPZ	●	●	●	●

5.2 PLD Programmers (Continued)

DATA I/O MODEL 29

The Data I/O Model 29 is one of the more popular first-generation programmers. It is equipped with a menu-driven control structure and most of the features useful for programming the more common PLDs. The Model 29 also programs PROMs and other special-purpose devices. An attachment called "Logic-Pack" is required, containing the basic circuitry needed for programming PLDs. In addition, various classes of PLDs require specific socket adapters to be added to the Logic-Pack. For example, when equipped with the 303A-001A adapter, it is capable of handling all NSC lateral-fuse PAL® and GAL® devices. It is connected to the software tool platform by means of an RS232C cable which runs fuse-map data in the form of JEDEC files across the link at 9600 Baud.

Files are down-loaded to the Model 29 through a number of steps, which include:

- Entering device family and pinout code
- Configuring the programmer
- Configuring the communications software in the platform
- Entering file name of data to be received
- Verifying through a checksum that the correct data was sent

Devices are then programmed with the down-loaded pattern by inserting a blank device of the correct type in the socket and selecting the Program function from the menu. Patterns can be copied from master devices, but only by alternating the devices in the single programming socket.

DATA I/O UNISITE

This is one of the most complex hardware tools. It is a stand-alone system which includes its own disk drives and can be considered a third generation system. The drives are used to input the Unisite system software (drive "A") and the algorithm software appropriate to the device (drive "B"). This allows a wide variety of devices to be handled from a single hardware socket. The system is connected to the platform by means of an RS232C cable and a communications software package, such as VTERM from Coefficient Systems must be installed and run.

The platform is then configured for the correct terminal type and communication configuration before the file transfer is set up. The Unisite is then powered on and configured through menu prompts with the following steps:

- Select terminal type
- Select device type
- Select device manufacturer/supplier
- Transfer and down-load data
- Select file name and protocol
- Select program function with options
- Insert device and perform programming

LOGICAL DEVICES ALLPRO

The ALLPRO from Logical Devices is one of the most popular second generation hardware tools. It is supplied with its own software and interface card, which must be installed into the standard card slot of an IBM-compatible PC. Once the software is running in the platform, the following steps need to be performed, using the menu prompts supplied:

- Select device
- Choose library and device
- Read a formatted file (download)

The device to be programmed must then be placed in the socket and the program function selected from the menu. Programming and sum checking are then performed automatically.

LOGICAL DEVICES PALPRO-2X

This system can be seen as a simpler version of the ALLPRO which does not have the same flexibility in the number of devices which can be handled.

The PALPRO does not involve installation of any cards, but simply connects to the software platform via the usual RS232C cable.

STAG MICROSYSTEMS PPZ

This is a full-featured programmer which is controlled using its own keyboard and CRT. It therefore requires no special software to be installed in the platform. It is connected through an RS232C line to the software platform. Down-loading is accomplished by placing the programmer into input mode and issuing a standard DOS COPY command to the PC. Remaining programming and test procedures are invoked directly through the built-in keyboard/CRT.

STAG MICROSYSTEMS ZL30A

The ZL30A is a simpler second generation device which handles most of the standard PLDs. It requires a few hardware configuration selections to be set up for communications with the platform. Hardware configuration algorithm for the type of device to be programmed is selected by software by insertion of the appropriate "smartcard" into a reader on the programmer.

Operation is similar to other programmers. Device selection is performed by entering special codes via the keyboard. The actual download and program functions are again performed by programmer controls.

5.3 National PLAN Software

The PLAN software package provided by National Semiconductor provides an interactive PLD design and development tool for digital system designers. For a detailed description of PLAN software, its installation and operation, refer to the PLAN documentation. PLAN equation syntax orginally evolved from the PALASM™ assembler from MMI, and is completely upward-compatible regarding logic-defining equations for devices common to both vendors. The PLAN package allows the system designer to begin with a text file specifying the circuit to be implemented, then provides support for the optimal device selection to accommodate the design among the available devices, assists in providing documentation, including a pin-list, and finally generates test vectors and format data to facilitate device programming and functional testing.

5.3 National PLAN Software (Continued)

FEATURES OF PLAN SOFTWARE

The main advantages to the designer of the PLAN software are:

- PLAN inputs may be from any standard editor or word processor
- PLAN equations are device-independent
- Automatic device selection
- Automatic pin-list generation
- Facilitates PAL-to-GAL JEDEC map conversion
- Reverse compilation of existing device maps
- Design documentation

PLAN software runs on an IBM®/PC or compatible platform, which must include:

- At least 512 kbytes of memory
- MS-DOS version 2.0 (or later)

PLAN SOFTWARE DESIGN ENTRY

The assembler is table-driven and permits selection among the industry-standard and proprietary products available from NSC. The range of the assembler input language is constantly being expanded as other devices are added to the product line without any changes to the language itself. This function-specific language is completely device-independent. This spares the designer from needing to understand the functional details of a larger number of specific PLD products.

PLAN software also allows truth table data entry. This is then converted to minimized Boolean equivalent equations. The format required for the Boolean input complies with accepted syntax to permit the use of third-party logic minimizers, logic simulators and vector generators.

DESIGN COMPILATION

PLAN software is unique in that it offers the designer an automatic selection of devices to contain the design once the specification equations have been entered. It will select the optimal device which is capable of implementing all the required functions and input/output requirements without intervention. Naturally, this can be edited by the designer, if desired.

The results of the assembly of the design are stored in a file called the JEDEC file, which is fully compatible with the current JEDEC 3A standard for use on all industry-standard device programmers.

The output from the assembler includes documentation of the design in addition to the fuse map output. This includes a graphic device pinout and a fuse map in readable form.

BACK TRANSLATION

The JED2BEQ program is part of the PLAN package. It allows existing JEDEC fuse maps to be reverse-compiled into the equations which describe it. This is extremely useful in analyzing the contents of a device for which there is no documentation or for building modifications to an existing design. The derived equations may then be targeted, via the PLAN assembler, to another compatible PLD.

PAL-TO-GAL MAP CONVERSION

Due to the more flexible generic nature of GAL devices, it is sometimes useful to be able to replace PAL devices in existing designs with equivalent GAL parts. The PAL2GAL program in the PLAN package is capable of taking any design from most standard PAL products and converting it into the equivalent specfification for a GAL device by creating a GAL JEDEC file.

To illustrate the operation of the PAL2GAL utility, let's run the JEDEC file previously created for the 6-bit shift register example in a PAL16R6 (*Figure 4-7)* to derive a new JEDEC file suitable for programming the same function using a GAL16V8. The resulting JEDEC map is shown in *Figure 5-1*.

```
PAL2GAL v1.14 Engineering Version
(C) National Semiconductor 1988  01/01/80   04:20:14
Source file name: A:SHIFTER.JED   Device: PAL16R6*
QP20* QF2194* F0*
L0000
01111111111111111111111111111011
11111110111111111111111111111111*
L0256
10111110111111111111111111111011
01111111111011111111111111111011
10101111111111111111111111110111
01111011111111111111111111110111*
L0512
10111111111011111111111111111011
01111111111111101111111111111011
10111110111111111111111111110111
01111111101111111111111111110111*
L0768
10111111111111101111111111111011
01111111111111111110111111111011
10111111111011111111111111110111
01111111111110111111111111110111*
L1024
10111111111111111110111111111011
01111111111111111111111011111011
10111111111111101111111111110111
01111111111111111011111111110111*
L1280
10111111111111111111111011111011
01111111111111111111111111101011
10111111111111111110111111110111
01111111111111111111101111110111*
L1536
10111111111111111111111111101011
01111111111111111111111111111010
10111111111111111111111011110111
01111111111111111111111110110111*
L1792
10111111111111111111111111110111
11111111111111111111111111101111*
L2048
00000000*
L2120
10000001*
L2128
11111111111111111111111111111111
11111111111111111111111111111111*
L2192
01*
C6FBE* FCF5
```

TL/L/9989–1

FIGURE 5-1. JEDEC File for 6-Bit Shift Register Example Converted from PAL16R6 to GAL16V8 Using PAL2GAL Utility Software

5.4 Other Assemblers/Compilers

PALASM SOFTWARE

This is a software package suppied by AMD/MMI. It is a PLD assembler which offers a number of software modules to assist in the implementation stage of PLD design. It is available in versions for the IBM PC and other computing platforms.

PALASM software accepts designs defined as either Boolean equations or state equations. It includes event-driven simulation and automatic logic reduction. It also has a JEDEC manipulator, which permits disassembly from an existing fuse plot and editing directly in the JEDEC file without having to go through the complete design process.

The implementation of the 6-bit shift register in PALASM software is most efficiently entered using the Boolean equations derived in Section 4.3. The resulting file, named SHIFTER.PDS is shown in *Figure 5-2*. Note that it also includes input test vectors for use by the simulator as created manually in *Figure 4-6*. Running the simulator will result in a set of output vectors which will be transferred to the programmer as part of the JEDEC file.

ABEL™ SOFTWARE

The ABEL package is an example of the second-generation PLD development tools which are usually referred to as high-level assemblers.

From an early version, which supported many PLD devices with logic reduction, simulation and generation of design documentation, it has become one of the standard software tools, capable of supporting a wide variety of PLDs, including PROMs. The most important feature which the most recent developments of ABEL software have added include:

- Revised simulation to support asynchronous devices and macrocells
- Syntax support of multiple-feedback paths
- Library of device-specific macros and functions
- JEDEC-to-ABEL conversion (for recovering undocumented designs)

ABEL software is produced by Data I/O and is available in versions for platforms like the IBM/PC, VAX™ and others. The design can be entered using any standard text editor. Any combination of Boolean equations, truth tables or state diagrams can be used. The description falls into four main sections:

- Declarations section, where sets are defined
- Equations section, where Boolean equations are entered
- Truth Table section, where functional tables are entered
- Test Vector section, where the behavior during simulation is given

ABEL automatically performs logic reduction, simulation and conversion to a JEDEC file without requiring further intervention unless some error is encountered.

CUPL ™ SOFTWARE

The CUPL package is a high-level compiler similar to ABEL software, which provides a number of functions not normally found in a standard PLD assembler. The input syntax is based on the C programming language. The high-level PLD support language in CUPL software permits development of designs using a systems approach. To this end, several features are supplied, including:

- self-documenting syntax
- state machine input option
- macro substitution
- flexible format
- use of symbolic names
- bit-field capability
- pre-processor functions
- output polarity selection

The macro substitution allows for considerable reduction in the number of keystrokes required for data entry, particularly for more complex functions. The pre-processor function allows the source file to be written in a much more generalized manner up until the actual compilation by the main assembler. This includes definition of the functions in state machine syntax and generalized arguments.

5.5 Pre-Assembler Software Tools

The additional functions, such as state-machine entry offered by many software tools, interact with the main assembly software through what is effectively the industry-standard Boolean equation interface. Taking advantage of this, several device-independent pre-processing software tools providing such functions have been marketed independently and can be used in conjunction with the standard Boolean-entry software tools. The range of pre-processing tools include:

- Logic minimizers
- State machine compilers
- Workstation graphic schematic capture compilers (e.g., Netlist)
- Test vector generators

Some high-level assemblers provide a selection of these functions, which are also device-independent. They can be used independently with other assemblers to provide that function. As an example, the logic minimizer in the CUPL package can be used to provide this function in conjunction with the PLAN assembler.

```
TITLE 6-BIT CASCADABLE SHIFT REGISTER EXAMPLE
PATTERN PALASM2 VERSION
REVISION REV. 0
AUTHOR APPLICATIONS ENGINEERING
COMPANY NATIONAL SEMICONDUCTOR
DATE MAY, 1988
CHIP SHIFTER PAL16R6
CLK SR D0 D1 D2 D3 D4 D5 SL GND
/G RILO Q5 Q4 Q3 Q2 Q1 Q0 LIRO VCC
EQUATIONS
/Q0 := /Q0*/SR*/SL + /Q1*SR*/SL + /LIRO*/SR*SL + /D0*SR*SL
/Q1 := /Q1*/SR*/SL + /Q2*SR*/SL + /Q0*/SR*SL + /D1*SR*SL
/Q2 := /Q2*/SR*/SL + /Q3*SR*/SL + /Q1*/SR*SL + /D2*SR*SL
/Q3 := /Q3*/SR*/SL + /Q4*SR*/SL + /Q2*/SR*SL + /D3*SR*SL
/Q4 := /Q4*/SR*/SL + /Q5*SR*/SL + /Q3*/SR*SL + /D4*SR*SL
/Q5 := /Q5*/SR*/SL + /RILO*SR*/SL + /Q4*/SR*SL + /D5*SR*SL
/LIRO = /Q0
LIRO.TRST = SR*/SL
/RILO = /Q5
RILO.TRST = /SR*SL
;
SIMULATION
SETF G /CLK
TRACE_ON SL SR D5 D4 D3 D2 D1 D0 LIRO RILO Q5 Q4 Q3 Q2 Q1 Q0
SETF SL SR /D5 /D4 /D3 /D2 /D1 /D0 LIRO RILO; LOAD ALL ZEROS
CLOCKF CLK
SETF /SL /SR D5 D4 D3 D2 D1 D0; HOLD ZEROS
CLOCKF CLK
SETF SL LIRO; SHIFT LEFT SINGLE 1
CLOCKF CLK
SETF /LIRO
FOR I:= 1 TO 6 DO BEGIN
CLOCKF CLK
END
SETF SL SR D5 D4 D3 D2 D1 D0 /LIRO /RILO; LOAD ALL ONES
CLOCKF CLK
SETF /SL /SR /D5 /D4 /D3 /D2 /D1 /D0; HOLD ONES
CLOCKF CLK
SETF SR /RILO; SHIFT RIGHT SINGLE 0
CLOCKF CLK
SETF RILO
FOR I:= 1 TO 6 DO BEGIN
CLOCKF CLK
END
TRACE_OFF
```

TL/L/9989-2

FIGURE 5-2. PALASM2 Input File for Shifter Example

Section 6

Fabrication of Programmable Logic

6

Section 6 Contents

6.1 PLD Technologies 6-3
- Standard, "Series-A" and "Series-B" TTL PAL devices
- ECL PAL Devices
- "Series-D" TTL PAL Devices with vertical fuses
- E^2CMOS GAL Devices

6.2 Quality and Factory Testing 6-5
- Product reliability
- Test circuitry
- Normal PLD factory test
- Pre-programmed PLD factory test
- NML factory test
- Customer handling and test

National
Semiconductor

Section 6
Fabrication of Programmable Logic

6.1 PLD Technologies

National Semiconductor is a broad-based supplier of programmable logic products. PLD's are offered in a wide range of circuit and programming technologies to address the diverse needs of most customer applications. PLD's can be seen as a stage of sophistication in the continuum from standard TTL logic to full-custom circuits. To provide the flexibility of configurable logic, PLD's make use of a number of circuit and programming technologies. These vary, depending on the type of PLD.

STANDARD, "SERIES-A" AND "SERIES-B" TTL PAL DEVICES

These are the earliest PLD families manufactured by National Semiconductor. The devices are fabricated using a traditional Schottky-TTL process similar to that used in the low-power Schottky (LS) TTL logic family of standard SSI/MSI functions. The addition of Titanium-Tungsten (TiW) fuse-links to the basic circuit technology allows the logic to be configured in the field after manufacturing is complete. This is the same programming technology as used in LS TTL PROM devices.

The fuse-links used in this programming technology are also referred to as "lateral fuses". The actual programming cell consists of a transistor with a fuse element in series with its emitter (as shown in *Figure 6-1*). It physically consists of a path, laid down during the metallization stage of the process, which is designed to pass a limited amount of current. When a higher current is passed, the metal heats to its melting point and surface tension draws the two sides of the fuse apart, leaving no electrical connection across the fuse.

While this is a reliable process, it is a one-time effect and therefore cannot be tested fully before the device is in the field. The first attempt to blow a fuse must be successful, otherwise the current passing through a partly-blown fuse will not be sufficient to blow it completely on successive attempts.

ECL PAL DEVICES

For ECL systems, high-speed PAL devices have been implemented with ECL I/Os using an oxide-isolated process known as OXISS. OXISS is a fully ion-implanted Schottky bipolar process with a 2 micron minimum feature size and two-layer metal interconnect. The ECL PAL products use the same Titanium-Tungsten lateral fuses as used in the original standard, Series-A and Series-B TTL PAL devices. The ECL PAL products are also available with both 10 KH and 100K compatibility.

More recent developments are based on National's ASPECT (Advanced Single-Poly Emitter-Coupled Technology) process. This is an oxide-isolated, self-aligned, contactless poly-emitter process which uses a single poly and two metal interconnect layers. The smaller geometries result in both lower gate power requirements and a higher device speed.

"SERIES-D" TTL PAL DEVICES WITH VERTICAL FUSES

National "Series-D" PAL devices, and faster TTL PAL devices being developed at National, use a new technology offering more compact geometries and higher overall device speeds than seen in the preceeding TTL PAL products. The technology used in Series-D devices is based on National Semiconductor's "FAST-Z" fully ion-implanted, isoplanar, Schottky-TTL process, similar to that used in the FAST® logic family.

Advances in programming technology have led to an improvement over the basic TiW lateral fuses used in the standard TTL PAL devices described earlier. This new technique uses a programming cell referred to as a "vertical fuse".

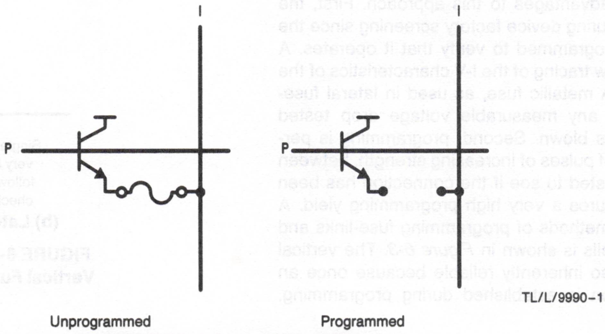
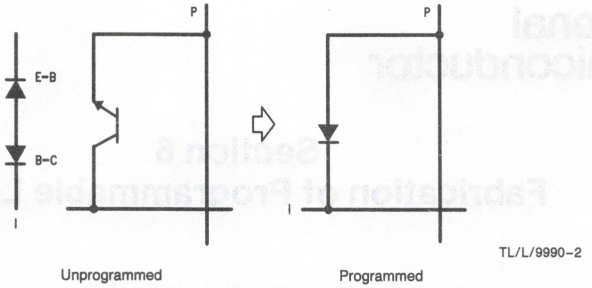
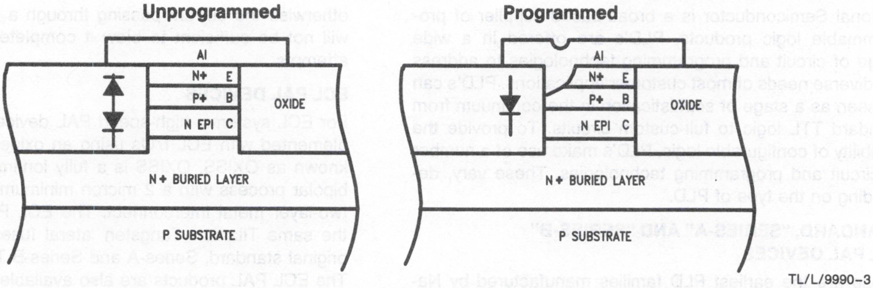


FIGURE 6-1. Lateral Fuse Circuit



(a) Equivalent Circuits



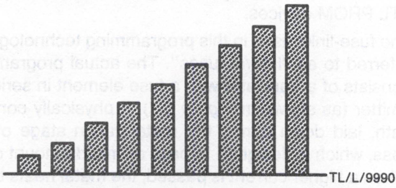
(b) Crossection

FIGURE 6-2. Vertical Fuse Circuit

Instead of a transistor and fuse-link in series, as shown in *Figure 6-1,* the vertical fuse cell consists of an open-base transistor connecting an input line (I) to a product line (P), as shown in *Figure 6-2.* Due to the high impedance of the transistor, the input line is not logically connected to the product line until the cell has been "programmed". Programming produces a short circuit through the emitter-base junction of the transistor establishing the connection. The initial state of a vertical fuse is therefore opposite that of the lateral fuse-link.

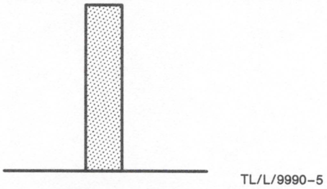
The programming mechanism is driven by Avalanche-Induced Migration (AIM). If the cell is considered a pair of back-to-back diodes, forcing a controlled current through the emitter of the cell eventually induces avalanche breakdown of the emitter-base junction. Heat generated locally causes the Aluminum/Silicon interface to reach eutectic temperature. This causes the Aluminum to diffuse through the emitter to the emitter-base junction, which causes a permanent short circuit.

There are several advantages to this approach. First, the cell can be tested during device factory screening since the cell need not be programmed to verify that it operates. A test current can allow tracing of the I-V characteristics of the connecting diode. A metallic fuse, as used in lateral fuse-links, cannot have any measurable voltage drop tested across it before it is blown. Second, programming is performed by a series of pulses of increasing strength, between which the fuse is tested to see if the connection has been made yet. This ensures a very high programming yield. A comparison of the methods of programming fuse-links and AIM vertical fuse cells is shown in *Figure 6-3.* The vertical fuse structure is also inherently reliable because once an adequate connection is established during programming, any degree of aluminum migration during continued operation can not adversely affect cell integrity.



Series of current pulses of increasing magnitude each followed by a read to check the fuse state

(a) Vertical Fuse



Single voltage pulse of very high magnitude followed by a read to check the fuse state

(b) Lateral Fuse-Link

FIGURE 6-3. Programming a Vertical Fuse vs Lateral Fuse

The programming algorithm for the vertical fuse devices in the programming equipment compensates for the opposite initial state of vertical fuses. Therefore the same JEDEC map as for all other standard PAL devices can be used without alteration.

E^{2}CMOS GAL DEVICES

Electrically-Erasable CMOS offers many advantages as a technology for PLDs. Most importantly, it offers the ability to erase and reprogram devices. This allows lower part usage, particularly at the development and prototype stages, by allowing the same device to be re-used or revised. This extends into manufacturing because the technology allows 100% factory testability without encountering the cycling difficulties or windowed packages associated with UV-based devices. 100% programming and functional yields are possible.

The power requirements of such a technology are very low when compared to standard bipolar. The technology has developed to a point where performance is comparable to standard PLDs, if not to the higher-speed parts. The low-power characteristics permit higher circuit density and higher reliability, which are particularly important in remote or power-conscious environments, such as in telecommunications.

The E^{2}CMOS technology is under further development to allow such devices to be programmed while in circuit, which will add even more to their flexibility and usefulness in manufacturing since they can be assembled in-circuit with all other components before being configured.

6.2 Quality and Factory Testing

PRODUCT RELIABILITY

National Semiconductor implements a reliability program for all of its integrated circuits. Reliability data is available for individual parts from the local sales office. It consists of:

- New product, package and process qualifications
- Existing product, package and process change qualifications
- Existing product, package and process monitoring

Product qualification testing performed by National meets or exceeds MIL standard 883. For a more thorough discussion of product quality and reliability, refer to the National Semiconductor Reliability Handbook.

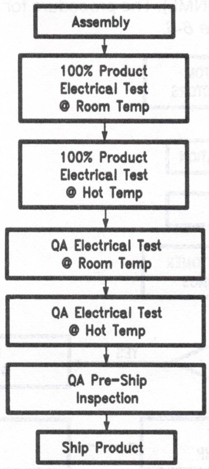
TEST CIRCUITRY

All standard bipolar and ECL PAL devices from National Semiconductor are fabricated with a number of test fuses and dedicated test circuitry as a part of the device. During final testing of each device, the functional paths, electrical integrity, cell programmability and the programming circuitry are all verified. The special test circuitry is accessed under non-operational modes during functional testing.

NORMAL PLD FACTORY TEST

Most commercial grade PLDs are available under the "A+" product enhancement program. This includes burn-in testing under bias and temperature cycling. For details of the "A+" program, refer to the "National Semiconductor A+/B+ Product Enhancement Programs" Brochure.

The standard procedure for unprogrammed PAL devices through National Semiconductor's factory is shown in *Figure 6-4*.

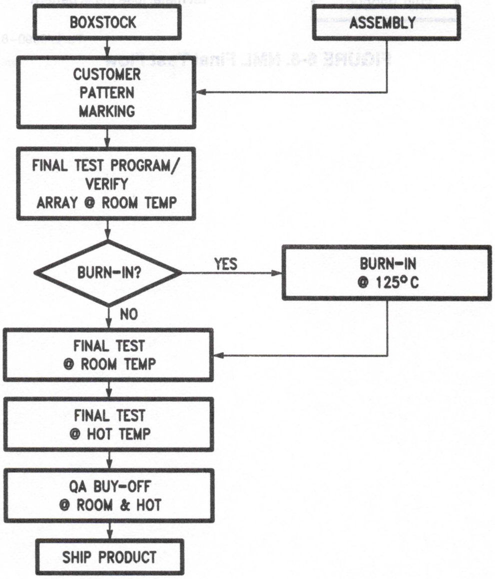

Assembly
100% Product Electrical Test @ Room Temp
100% Product Electrical Test @ Hot Temp
QA Electrical Test @ Room Temp
QA Electrical Test @ Hot Temp
QA Pre-Ship Inspection
Ship Product

TL/L/9990-6

FIGURE 6-4. Final Test Flow for Unprogrammed PLD

PRE-PROGRAMMED PLD FACTORY TEST

As discussed earlier, National provides the service of providing devices already configured with fuse map configuration to customers in a fully-tested state. The procedure for doing this is shown in *Figure 6-5*.



TL/L/9990-7

FIGURE 6-5. Pre-Programmed PLD Final Test Flow

NML FACTORY TEST

For volume production, National Semiconductor offers the mask-option approach to PLD production, known as National Masked Logic (NML). The procedure for the NML process is shown in *Figure 6-6*.

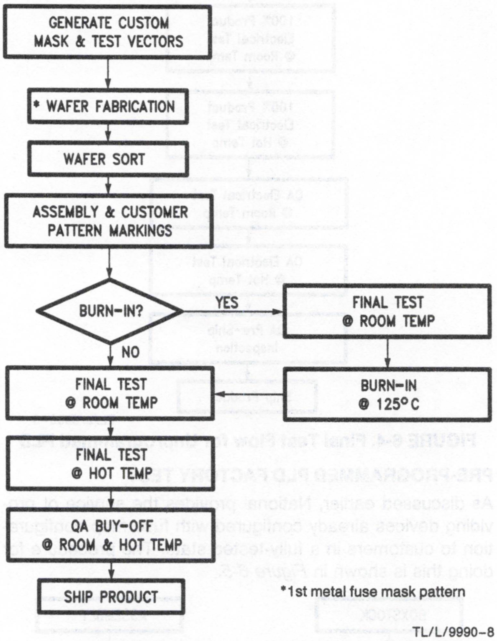

GENERATE CUSTOM MASK & TEST VECTORS
* WAFER FABRICATION
WAFER SORT
ASSEMBLY & CUSTOMER PATTERN MARKINGS
BURN-IN?
YES
NO
FINAL TEST @ ROOM TEMP
BURN-IN @ 125°C
FINAL TEST @ ROOM TEMP
FINAL TEST @ HOT TEMP
QA BUY-OFF @ ROOM & HOT TEMP
SHIP PRODUCT
*1st metal fuse mask pattern
TL/L/9990-8

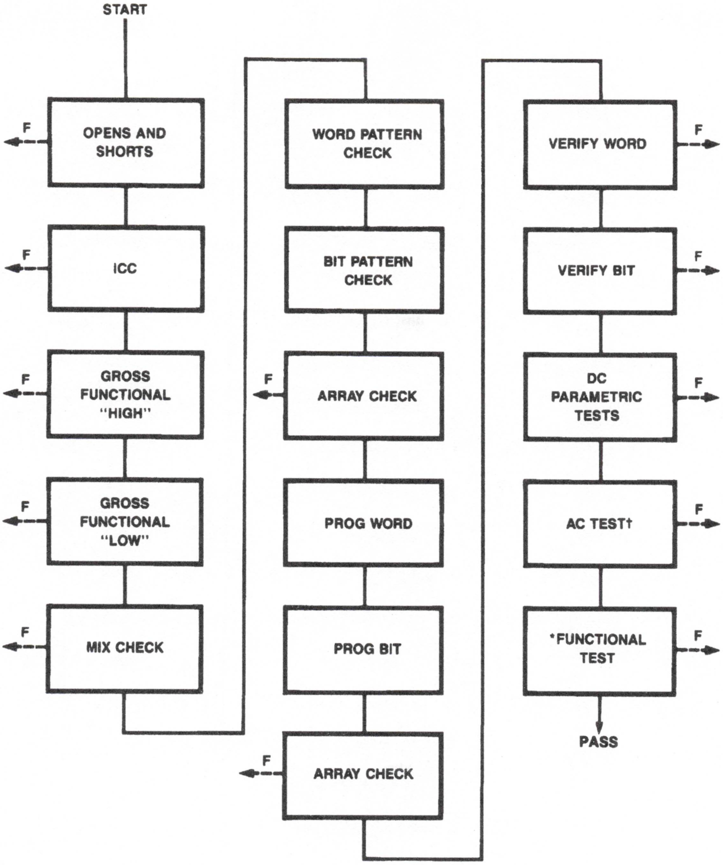
FIGURE 6-6. NML Final Test Flow

CUSTOMER HANDLING AND TEST

PLD devices are more sensitive to handling than most other integrated circuits. Care should be exercised to prevent static buildup when handling parts, particularly CMOS devices. Handling is the primary cause of defective material. To minimize the problems with production, the number of steps during which parts are handled should be kept to an absolute minimum.

A variety of approaches on device test are available to the manufacturer of systems using PLDs. For smaller production quantities, the programming hardware used for development described in Section 5 may offer an adequate solution. As quantities become larger, some programmer vendors offer add-on auto-handler packages which can handle devices in volume.

For more extensive manufacturing flows, programs are available for most of the common IC tester systems which allow the tester to perform device programming as a part of normal device flow. Other than this, National offers pre-programmed devices as described earlier.


START
OPENS AND SHORTS
ICC
GROSS FUNCTIONAL "HIGH"
GROSS FUNCTIONAL "LOW"
MIX CHECK
WORD PATTERN CHECK
BIT PATTERN CHECK
ARRAY CHECK
PROG WORD
PROG BIT
ARRAY CHECK
VERIFY WORD
VERIFY BIT
DC PARAMETRIC TESTS
AC TEST†
*FUNCTIONAL TEST
PASS
F

TL/L/9990–9

†For sample only

*For NML/PROGRAMMED PAL

FIGURE 6-7. PAL Device Electrical Test Sequence

Section 7

Application Examples

Section 7 Contents

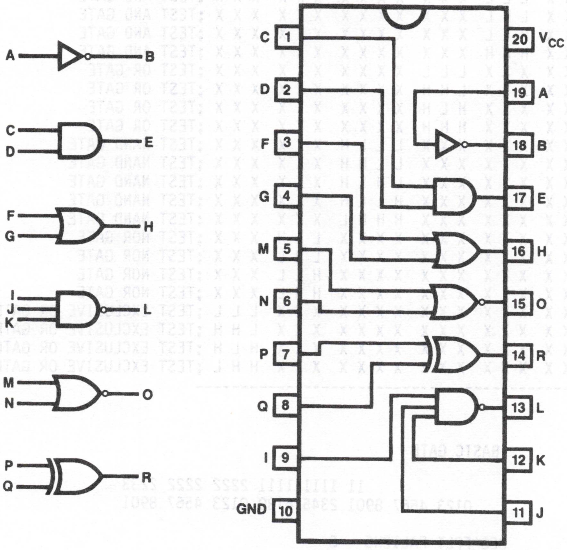
7.1 Basic Gates (PAL12H6) 7-3
7.2 Basic Flip-Flops (GAL16V8) 7-7
7.3 Memory-Mapped I/O Address Decoder (PAL16L2) 7-12
7.4 Quad 4-to-1 Multiplexer (GAL20V8) 7-15
7.5 Dual 8-to-1 Multiplexer (PAL20L2) 7-19
7.6 16-to-1 Multiplexer (PAL20C1) 7-22
7.7 4-Bit Counter with 2 Parallel-Load Ports (PAL16R4) 7-25
7.8 7-Bit Counter with Parallel-Load (GAL20V8) 7-28
7.9 8-Bit Counter with Parallel-Load (PAL20X8) 7-32
7.10 10-Bit Up/Down Counter (PAL20X10) 7-33
7.11 8-Bit Cascadable Shift Register (PAL20X8) 7-37
7.12 8-Bit Equality Comparator (PAL16C1) 7-40
7.13 8-Bit Barrel Shifter (GAL20V8) 7-41
7.14 8-Input Priority Encoder (GAL16V8) 7-45
7.15 16-Input Priority Encoder (GAL20V8) 7-49
7.16 Hexadecimal 7-Segment Display Encoder (PAL16L8) 7-53
7.17 Dual-Port RAM Controller (GAL16V8) 7-58
7.18 8086 CPU Board Random Control Logic (PAL12H6) 7-67
7.19 DP84312 DRAM Controller Interface for NS32032 (PAL16R6) 7-69
7.20 DP84322 DRAM Controller Interface for 68000 CPU (PAL16R4) 7-77
7.21 DP84332 DRAM Controller Interface for 8086/8088 (PAL16R8) 7-94
7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032 7-104

National Semiconductor

Section 7
Application Examples*

7.1 Basic Gates

This example demonstrates how fusable logic can implement the basic inverter, AND OR, NAND, NOR and exclusive-OR functions. The PAL12H6 is selected because it has 12 inputs and 6 outputs.


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
C 1
D 2
F 3
G 4
M 5
N 6
P 7
Q 8
I 9
GND 10
20 V_{CC}
19 A
18 B
17 E
16 H
15 O
14 R
13 L
12 K
11 J

TL/L/9991-1

FIGURE 7.1.1. Basic Gates

---

*Applications contained in this section are for illustration purposes only and National makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

7.1 Basic Gates (Continued)

```
PALASM VERSION 1.5

PAL12H6
APPLICATIONS
BASIC GATE
NSC SANTA CLARA
C D F G M N P Q I GND J K L R O H E B A VCC
B = /A
E = C*D
H = F + G
L = /I + /J + /K
O = /M*/N
R = P*/Q + /P*Q
FUNCTION TABLE
A B C D E F G H I J K L M N O P Q R
-----------------------------------------
L H  X X X  X X X  X X X X  X X X  X X X ;TEST INVERTER
H L  X X X  X X X  X X X X  X X X  X X X ;TEST INVERTER
X X  L L L  X X X  X X X X  X X X  X X X ;TEST AND GATE
X X  L H L  X X X  X X X X  X X X  X X X ;TEST AND GATE
X X  H L L  X X X  X X X X  X X X  X X X ;TEST AND GATE
X X  H H H  X X X  X X X X  X X X  X X X ;TEST AND GATE
X X  X X X  L L L  X X X X  X X X  X X X ;TEST OR GATE
X X  X X X  L H H  X X X X  X X X  X X X ;TEST OR GATE
X X  X X X  H L H  X X X X  X X X  X X X ;TEST OR GATE
X X  X X X  H H H  X X X X  X X X  X X X ;TEST OR GATE
X X  X X X  X X X  L L L H  X X X  X X X ;TEST NAND GATE
X X  X X X  X X X  L L H H  X X X  X X X ;TEST NAND GATE
X X  X X X  X X X  L H L H  X X X  X X X ;TEST NAND GATE
X X  X X X  X X X  H L L H  X X X  X X X ;TEST NAND GATE
X X  X X X  X X X  H H H L  X X X  X X X ;TEST NAND GATE
X X  X X X  X X X  X X X X  L L H  X X X ;TEST NOR GATE
X X  X X X  X X X  X X X X  L H L  X X X ;TEST NOR GATE
X X  X X X  X X X  X X X X  H L L  X X X ;TEST NOR GATE
X X  X X X  X X X  X X X X  H H L  X X X ;TEST NOR GATE
X X  X X X  X X X  X X X X  X X X  L L L ;TEST EXCLUSIVE OR GATE
X X  X X X  X X X  X X X X  X X X  L H H ;TEST EXCLUSIVE OR GATE
X X  X X X  X X X  X X X X  X X X  H L H ;TEST EXCLUSIVE OR GATE
X X  X X X  X X X  X X X X  X X X  H H L ;TEST EXCLUSIVE OR GATE
------------------------------------------
```

TL/L/9991-2

```
      BASIC GATE

                      11 1111 1111 2222 2222 2233
        0123 4567 8901 2345 6789 0123 4567 8901

     BEG*FPLT PAL12H6    8

        8 ---- ---X --00 --00 --00 --00 ---- ---- /A
        9 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX
       10 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX
       11 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

       16 X-X- ---- --00 --00 --00 --00 ---- ---- C*D
       17 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

       24 ---- X--- --00 --00 --00 --00 ---- ---- F
       25 ---- ---- X-00 --00 --00 --00 ---- ---- G

       32 ---- ---- --00 -X00 -X00 --00 ---- ---- /M*/N
       33 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

       40 ---- ---- --00 --00 --00 X-00 -X-- ---- P*/Q
       41 ---- ---- --00 --00 --00 -X00 X--- ---- /P*Q

       48 ---- ---- --00 --00 --00 --00 ---- -X-- /I
       49 ---- ---- --00 --00 --00 --00 ---- ---X /J
       50 ---- ---- --00 --00 --00 --00 ---X ---- /K
       51 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

END*FPLT

 LEGEND:  X : FUSE NOT BLOWN (L,N,0)    - : FUSE BLOWN   (H,P,1)
          0 : PHANTOM FUSE   (L,N,0)    0 : PHANTOM FUSE (H,P,1)

 NUMBER OF FUSES BLOWN =  306
```

TL/L/9991-5

7.1 Basic Gates (Continued)

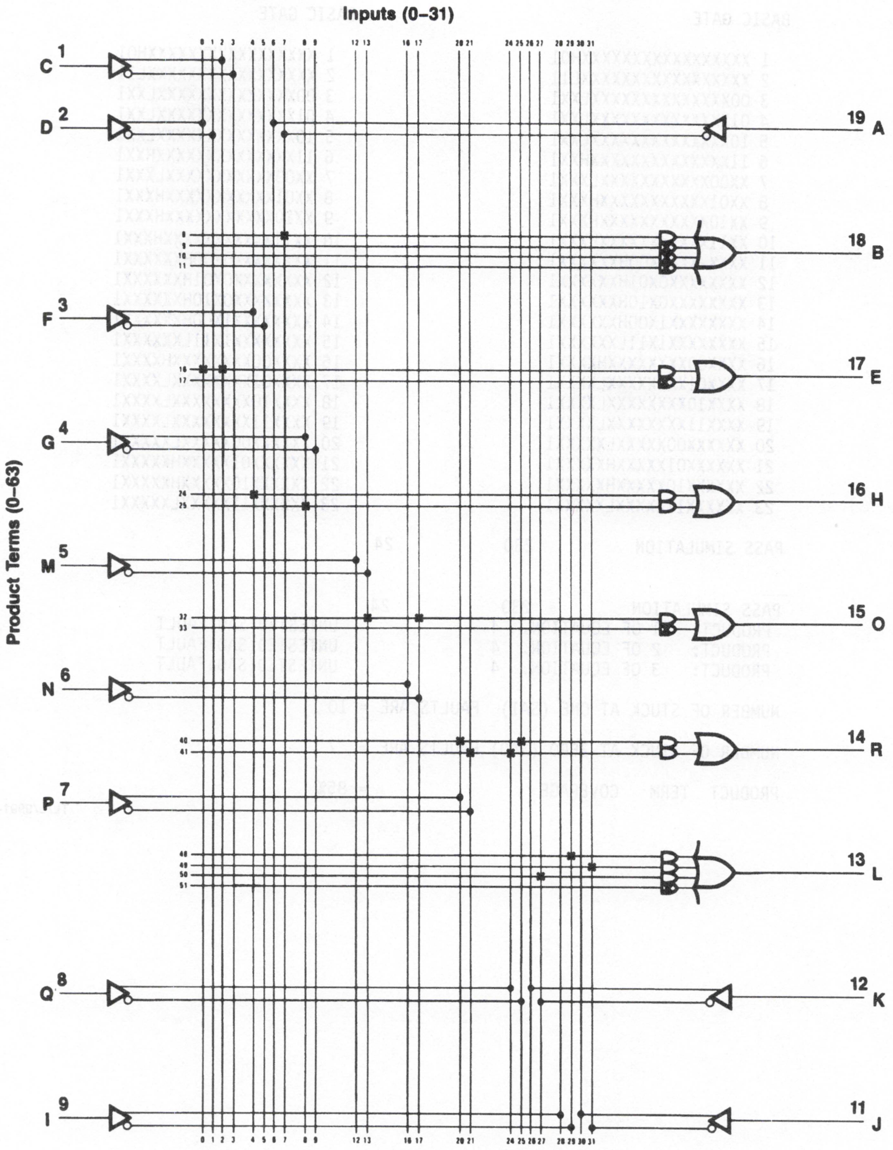
```
BASIC GATE

 1 XXXXXXXXXXXXXXXXXH01
 2 XXXXXXXXXXXXXXXXXL11
 3 00XXXXXXXXXXXXXXLXX1
 4 01XXXXXXXXXXXXXXLXX1
 5 10XXXXXXXXXXXXXXLXX1
 6 11XXXXXXXXXXXXXXHXX1
 7 XX00XXXXXXXXXXXLXXX1
 8 XX01XXXXXXXXXXXHXXX1
 9 XX10XXXXXXXXXXXHXXX1
10 XX11XXXXXXXXXXXHXXX1
11 XXXXXXXX0X00HXXXXXX1
12 XXXXXXXX0X01HXXXXXX1
13 XXXXXXXX0X10HXXXXXX1
14 XXXXXXXX1X00HXXXXXX1
15 XXXXXXXX1X11LXXXXXX1
16 XXXX00XXXXXXXXHXXXX1
17 XXXX01XXXXXXXXLXXXX1
18 XXXX10XXXXXXXXLXXXX1
19 XXXX11XXXXXXXXLXXXX1
20 XXXXXX00XXXXXLXXXXX1
21 XXXXXX01XXXXXHXXXXX1
22 XXXXXX10XXXXXHXXXXX1
23 XXXXXX11XXXXXLXXXXX1
```

```
BASIC GATE

 1 XXXXXXXXXXXXXXXXXH01
 2 XXXXXXXXXXXXXXXXXL11
 3 00XXXXXXXXXXXXXXLXX1
 4 01XXXXXXXXXXXXXXLXX1
 5 10XXXXXXXXXXXXXXLXX1
 6 11XXXXXXXXXXXXXXHXX1
 7 XX00XXXXXXXXXXXLXXX1
 8 XX01XXXXXXXXXXXHXXX1
 9 XX10XXXXXXXXXXXHXXX1
10 XX11XXXXXXXXXXXHXXX1
11 XXXXXXXX0X00HXXXXXX1
12 XXXXXXXX0X01HXXXXXX1
13 XXXXXXXX0X10HXXXXXX1
14 XXXXXXXX1X00HXXXXXX1
15 XXXXXXXX1X11LXXXXXX1
16 XXXX00XXXXXXXXHXXXX1
17 XXXX01XXXXXXXXLXXXX1
18 XXXX10XXXXXXXXLXXXX1
19 XXXX11XXXXXXXXLXXXX1
20 XXXXXX00XXXXXLXXXXX1
21 XXXXXX01XXXXXHXXXXX1
22 XXXXXX10XXXXXHXXXXX1
23 XXXXXX11XXXXXLXXXXX1
```

```
PASS SIMULATION          230            24


PASS SIMULATION          230            24
 PRODUCT:    1 OF EQUATION.  4                UNTESTED(SA0)FAULT
 PRODUCT:    2 OF EQUATION.  4                UNTESTED(SA0)FAULT
 PRODUCT:    3 OF EQUATION.  4                UNTESTED(SA0)FAULT

NUMBER OF STUCK AT ONE (SA1)  FAULTS ARE = 10

NUMBER OF STUCK AT ZERO (SA0) FAULTS ARE =  7

PRODUCT  TERM   COVERAGE                 = 85%
```

TL/L/9991-6

7.1 Basic Gates (Continued)


Inputs (0–31)
Product Terms (0–63)
0 1 2 3 4 5 6 7 8 9 12 13 16 17 20 21 24 25 26 27 28 29 30 31
1 C
2 D
3 F
4 G
5 M
6 N
7 P
8 Q
9 I
8
9
10
11
16
17
24
25
32
33
40
41
48
49
50
51
19 A
18 B
17 E
16 H
15 O
14 R
13 L
12 K
11 J

TL/L/9991–7

FIGURE 7.1.2. PAL12H6 Logic Diagram Showing Fuse Pattern of Basic Gates Example

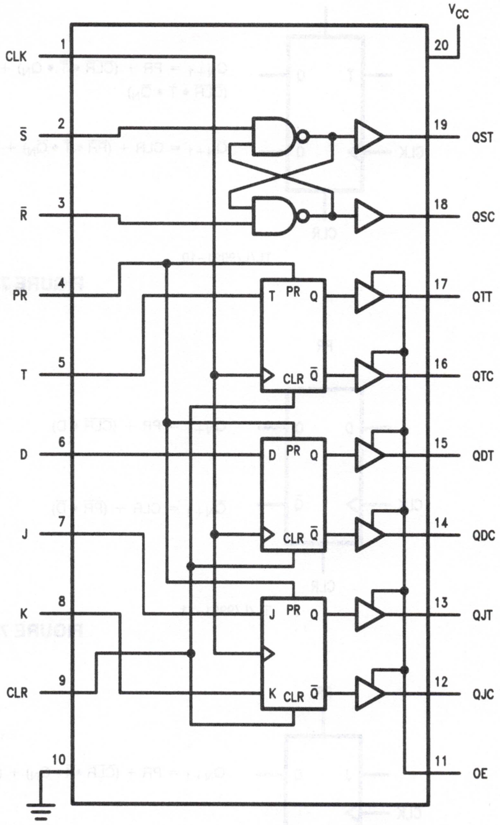
7.2 Basic Flip-Flops

DESCRIPTION

In the Basic Gates application on the preceding pages, each 'gate' was directly connected to an output pin. Here, the output registers of the GAL16V8 are used. A simple RS latch, a T (Toggle) flip-flop, a D flip-flop, and a JK flip-flop are incorporated into a GAL16V8 *(Figure 7.2.1)*. Each is shown with its truth table and defining equations in *Figures 7.2.2–7.2.5*. Note that all 3 flip-flops have synchronous preset (PR) and clear (CLR) inputs, while the RS latch does not. Also, the RS latch is not connected to the clock input; this was done to show the versatility of the GAL16V8—with a GAL device, the user is not locked in to a specific architecture.

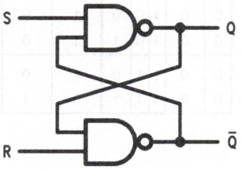
The CUPL design input file *(Figure 7.2.6)* and simulation file *(Figure 7.2.7)* are constructed by the designer. Each output must be given a distinct name, and any clocked circuit must be denoted with an appropriate extension (.D) in the logic equations. The simulation file is again provided for design verification.

This example has some subtle requirements that may not be apparent to the first-time user. When the RS latch is not being tested, it must remain in its latched state with the output levels specified, or with the variable N (not tested) specified instead. Also, when executing a preset or clear, remember that it will affect all flip-flops; even those not being tested will still respond. Finally, all output levels should be specified or marked with the variable N; the variable X, which indicates a 'don't care' condition, will not suffice.



TL/L/9991-8

FIGURE 7.2.1. Basic Flip-Flops Pinout



$$Q_{N+1} = \overline{S} + (R \bullet Q_N)$$

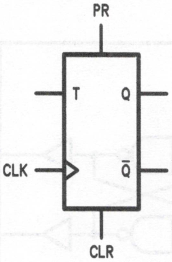
$$\overline{Q}_{N+1} = \overline{R} + (S \bullet \overline{Q}_N)$$

TL/L/9991-9

S	R	Q_N	Q_{N+1}	$\overline{Q}_{N+1}$	Comments
0	0	0	1	1	Invalid
0	0	1	1	1	
0	1	0	1	0	Set
0	1	1	1	0	
1	0	0	0	1	Reset
1	0	1	0	1	
1	1	0	0	1	Latch
1	1	1	1	0	

FIGURE 7.2.2. RS Latch

7.2 Basic Flip-Flops (Continued)



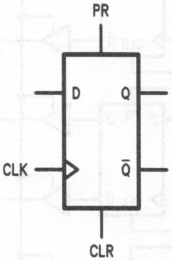
$$Q_{N+1} = PR + (\overline{CLR} \bullet \overline{T} \bullet Q_N) + (\overline{CLR} \bullet T \bullet \overline{Q}_N)$$

$$\overline{Q}_{N+1} = CLR + (\overline{PR} \bullet \overline{T} \bullet \overline{Q}_N) + (\overline{PR} \bullet T \bullet Q_N)$$

PR	CLR	T	Q_N	Q_{N+1}	$\overline{Q}_{N+1}$	Comments
1	1	X	X	1	1	Invalid
1	0	X	X	1	0	Preset
0	1	X	X	0	1	Clear
0	0	0	0	0	1	Hold
0	0	0	1	1	0	
0	0	1	0	1	0	Toggle
0	0	1	1	0	1	

TL/L/9991-10

FIGURE 7.2.3. T Flip-Flop



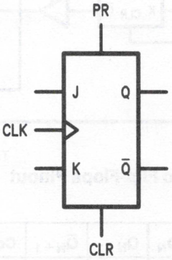
$$Q_{N+1} = PR + (\overline{CLR} \bullet D)$$

$$\overline{Q}_{N+1} = CLR + (\overline{PR} \bullet \overline{D})$$

PR	CLR	D	Q_N	Q_{N+1}	$\overline{Q}_{N+1}$	Comments
1	1	X	X	1	1	Invalid
1	0	X	X	1	0	Preset
0	1	X	X	0	1	Clear
0	0	0	0	0	1	Reset
0	0	0	1	0	1	
0	0	1	0	1	0	Set
0	0	1	1	1	0	

TL/L/9991-11

FIGURE 7.2.4. D Flip-Flop



$$Q_{N+1} = PR + (\overline{CLR} \bullet \overline{K} \bullet Q_N) + (\overline{CLR} \bullet J \bullet \overline{Q}_N)$$

$$\overline{Q}_{N+1} = CLR + (\overline{PR} \bullet \overline{J} \bullet \overline{Q}_N) + (\overline{PR} \bullet K \bullet Q_N)$$

PR	CLR	J	K	Q_N	Q_{N+1}	$\overline{Q}_{N+1}$	Comments
1	1	X	X	X	1	1	Invalid
1	0	X	X	X	1	0	Preset
0	1	X	X	X	0	1	Clear
0	0	0	0	0	0	1	Hold
0	0	0	0	1	1	0	
0	0	0	1	0	0	1	Reset
0	0	0	1	1	0	1	
0	0	1	0	0	1	0	Set
0	0	1	0	1	1	0	
0	0	1	1	0	1	0	Toggle
0	0	1	1	1	0	1	

TL/L/9991-12

FIGURE 7.2.5. JK Flip-Flop

```
/*****************************************************************/
/*                                                               */
/*                      CUPL INPUT FILE                          */
/*      Flip-flops and latches implemented in a GAL16V8          */
/*                                                               */
/*****************************************************************/
        PARTNO          456STX;
        NAME            FLIPFLOP;
        REV             1;
        DATE            4/11/86;
        DESIGNER        Joe Engineer;
        COMPANY         National Semiconductor;
        ASSEMBLY        Clock Board;
        LOCATION        U238;

/* RS latch */

        pin [2,3,19,18] = [S,R,QST,QSC];

/* T flip-flop */

        pin [5,17,16] = [T,QTT,QTC];

/* D flip-flop */

        pin [6,15,14] = [D,QDT,QDC];

/* JK flip-flop */

        pin [7,8,13,12] = [J,K,QJT,QJC];

/* control */

        pin [1,4,9,11] = [CLK,PR,CLR,OE];

/* logic equations */

        /* RS latch */

                QST = !S # (R & QST);
                QSC = !R # (S & QSC);

        /* T flip-flop */

                QTT.D = PR # (!CLR & !T & QTT) # (!CLR & T & QTC);
                QTC.D = CLR # (!PR & !T & QTC) # (!PR & T & QTT);

        /* D flip-flop */

                QDT.D = PR # (D & !CLR);
                QDC.D = CLR # (!D & !PR);

        /* JK flip-flop */

                QJT.D = PR # (J & QJC & !CLR) # (!K & QJT & !CLR);
                QJC.D = CLR # (!J & QJC & !PR) # (K & QJT & !PR);
```

TL/L/9991-15

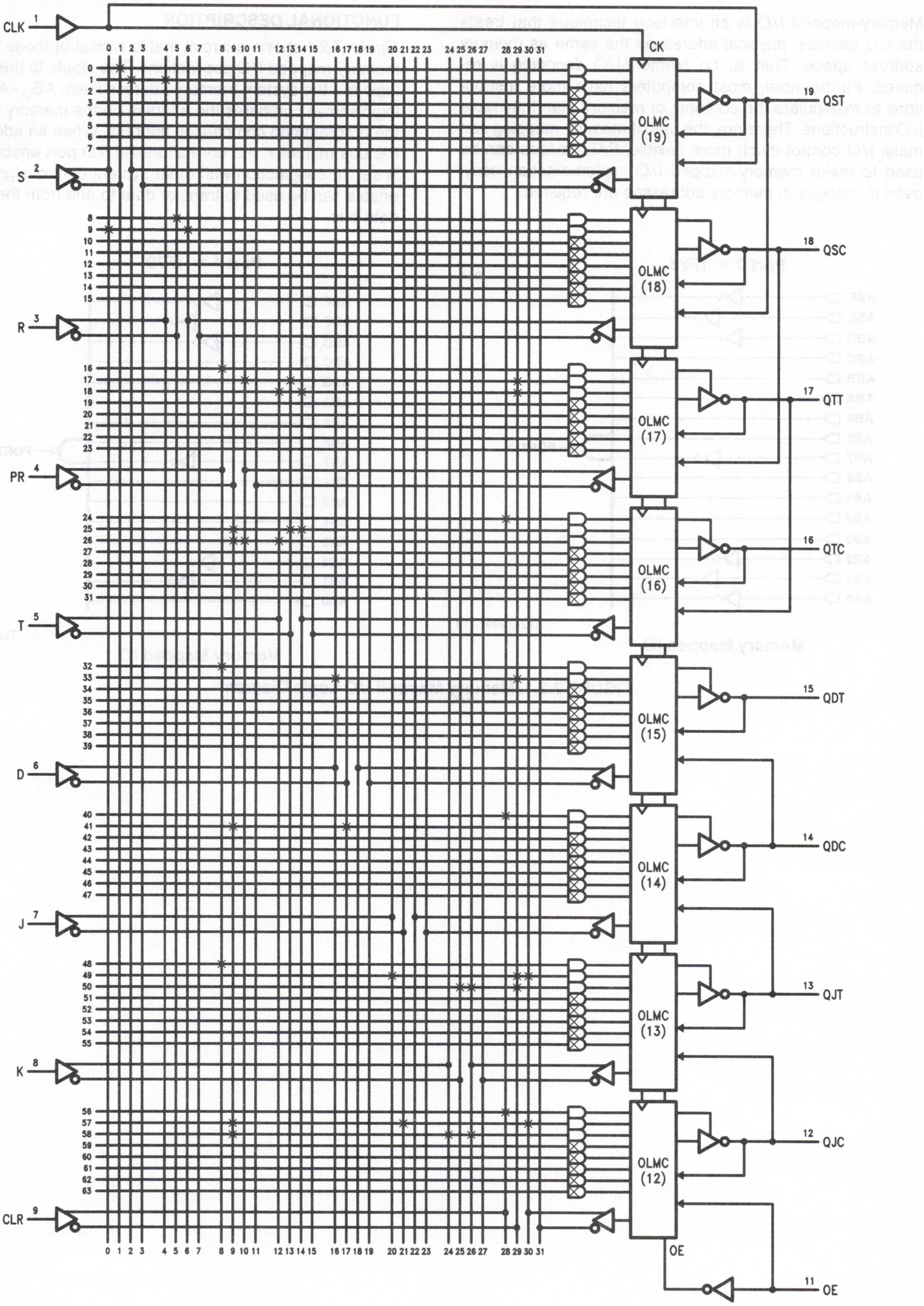
FIGURE 7.2.6. CUPL Input File

```
/******************************************************************/
/*                                                                */
/*                    CUPL SIMULATION FILE                        */
/*     Flip-flops and latches implemented in a GAL16V8            */
/*                                                                */
/******************************************************************/
        PARTNO          456STX;
        NAME            FLIPFLOP;
        REV             1;
        DATE            4/11/86;
        DESIGNER        Joe Engineer;
        COMPANY         National Semiconductor;
        ASSEMBLY        Clock Board;
        LOCATION        U238;

/* The Order statement specifies the layout of the vector table.
        %n = n spaces inserted between variables.                  */

order:  OE,%1,CLK,%2,S,R,%1,QST,QSC,%2,PR,%1,CLR,%2,
        T,%1,QTT,QTC,%2,D,%1,QDT,QDC,%2,J,K,%1,QJT,QJC;

vectors:
/*         RS-latch              T-FF      D-FF       JK-FF     */
/*OE CLK   SR QSTQSC   PR CLR   T QTTQTC  D QDTQDC   JK QJTQJC  */
   0  X    01  H  L     X  X    X  X  X   X  X  X    XX  X  X  /* set */
   0  X    10  L  H     X  X    X  X  X   X  X  X    XX  X  X  /* reset */
   0  X    11  L  H     X  X    X  X  X   X  X  X    XX  X  X  /* latch */
   0  X    10  L  H     X  X    X  X  X   X  X  X    XX  X  X  /* reset */
   0  X    01  H  L     X  X    X  X  X   X  X  X    XX  X  X  /* set */
   0  X    11  H  L     X  X    X  X  X   X  X  X    XX  X  X  /* latch */

   0  C    11  N  N     1  0    X  H  L   X  N  N    XX  N  N  /* preset */
   0  C    11  N  N     0  1    X  L  H   X  N  N    XX  N  N  /* clear */
   0  C    11  N  N     0  0    0  L  H   X  X  X    XX  X  X  /* hold   */
   0  C    11  N  N     0  0    1  H  L   X  X  X    XX  X  X  /* toggle */
   0  C    11  N  N     0  0    0  H  L   X  X  X    XX  X  X  /* hold   */
   0  C    11  N  N     0  0    1  L  H   X  X  X    XX  X  X  /* toggle */
   0  C    11  N  N     0  0    1  H  L   X  X  X    XX  X  X  /* toggle */
   0  C    11  N  N     1  0    X  N  N   X  H  L    XX  N  N  /* preset */
   0  C    11  N  N     0  1    X  N  N   X  L  H    XX  N  N  /* clear */
   0  C    11  N  N     0  0    X  X  X   0  L  H    XX  X  X
   0  C    11  N  N     0  0    X  X  X   1  H  L    XX  X  X  /* test */
   0  C    11  N  N     0  0    X  X  X   1  H  L    XX  X  X
   0  C    11  N  N     0  0    X  X  X   0  L  H    XX  X  X

   0  C    11  N  N     1  0    X  N  N   X  N  N    XX  H  L  /* preset */
   0  C    11  N  N     0  1    X  N  N   X  N  N    XX  L  H  /* clear */
   0  C    11  N  N     0  0    X  X  X   X  X  X    01  L  H
   0  C    11  N  N     0  0    X  X  X   X  X  X    00  L  H  /* hold */
   0  C    11  N  N     0  0    X  X  X   X  X  X    11  H  L  /* toggle */
   0  C    11  N  N     0  0    X  X  X   X  X  X    10  H  L
   0  C    11  N  N     0  0    X  X  X   X  X  X    00  H  L  /* hold */
   0  C    11  N  N     0  0    X  X  X   X  X  X    11  L  H  /* toggle */
   0  C    11  N  N     0  0    X  X  X   X  X  X    10  H  L
   0  C    11  N  N     0  0    X  X  X   X  X  X    01  L  H
```

TL/L/9991–16

FIGURE 7.2.7. CUPL Simulation File

7.2 Basic Flip-Flops (Continued)


CLK 1
S 2
R 3
PR 4
T 5
D 6
J 7
K 8
CLR 9
CK
OLMC (19)
OLMC (18)
OLMC (17)
OLMC (16)
OLMC (15)
OLMC (14)
OLMC (13)
OLMC (12)
OE
19 QST
18 QSC
17 QTT
16 QTC
15 QDT
14 QDC
13 QJT
12 QJC
11 OE
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

TL/L/9991-17

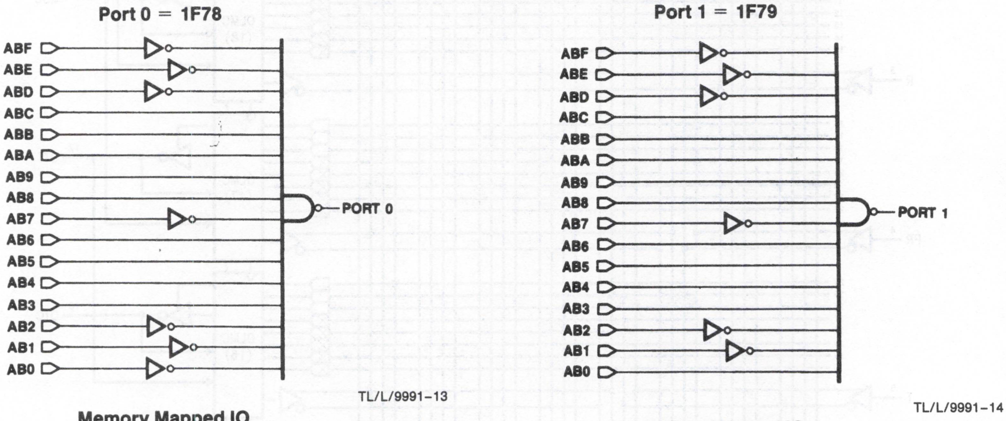
FIGURE 7.2.8. GAL16V8 Logic Diagram Showing Basic Flip-Flops Pattern

7.3 Memory-Mapped I/O Address Decoder

Memory-mapped I/O is an interface technique that treats the I/O devices' physical addresses the same as memory address space. That is, no Memory-I/O decoding is required. Furthermore, most computers have more instructions to manipulate the contents of memory than they have I/O instructions. Therefore, the use of memory mapping can make I/O control much more flexible. PAL devices can be used to make memory-mapped I/O implementation easy, even if changes in memory addresses are required.

FUNCTIONAL DESCRIPTION

Figure 7.3.1 shows a circuit that is typical of those found in memory-mapped I/O applications. The inputs to the decode logic are the system memory address lines, AB_O-AB_F. The logic shown compares the address on the memory bus with the programmed comparison address. When an address on the bus matches, the corresponding I/O port enable signal is set. In conjunction with other system control signals, this enable can be used to transfer data to and from the system data bus.


Port 0 = 1F78
ABF
ABE
ABD
ABC
ABB
ABA
AB9
AB8
AB7
AB6
AB5
AB4
AB3
AB2
AB1
AB0
PORT 0
TL/L/9991-13
Port 1 = 1F79
ABF
ABE
ABD
ABC
ABB
ABA
AB9
AB8
AB7
AB6
AB5
AB4
AB3
AB2
AB1
AB0
PORT 1
TL/L/9991-14

Memory Mapped IO

Memory Mapped IO

FIGURE 7.3.1. Memory Mapped I/O Logic Diagram

7.3 Memory-Mapped I/O Address Decoder (Continued)

PAL® DEVICE DESIGN

One PAL16L2 can be used to monitor a 16-bit address bus, fully decode addresses, and furnish enables to two ports, each of which can be anywhere within 64k of address space. Partial decoding for a larger number of ports can be done using other members of the PAL device family.

Typical logic equations for the memory-mapped I/O logic are as follows:

Port 0 = /AB0•/AB1•/AB2•AB3•AB4•AB5•AB6•/AB7•AB8•AB9•ABA•ABB•ABC•/ABD•/ABE•/ABF

Port 1 = AB0•/AB1•/AB2•AB3•AB4•AB5•AB6•/AB7•AB8•AB9•ABA•ABB•ABC•/ABD•/ABE•/ABF

The above example shows address decoding for memory locations IF78$_{H}$ and IF79$_{H}$. The equation terms could be changed to accommodate any 16-bit address.

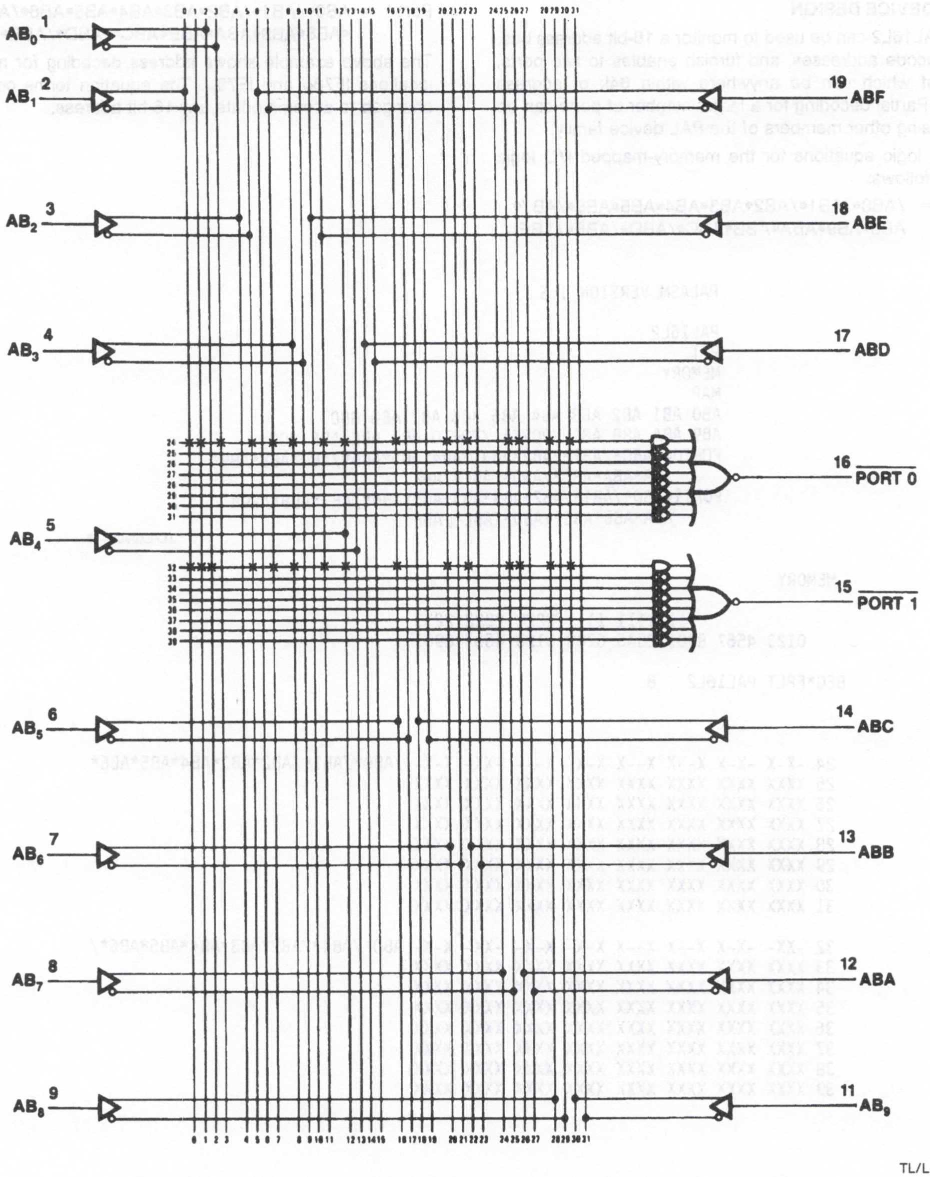
```
PALASM VERSION 1.5

PAL16L2
PAT
MEMORY
MAP
AB0 AB1 AB2 AB3 AB4 AB5 AB6 AB7 AB8 GND
AB9 ABA ABB ABC /PORT1 /PORT0 ABD ABE ABF VCC
PORT0=/AB0*/AB1*/AB2*AB3 *AB4*AB5*AB6*/AB7*AB8*AB9*
    ABA*ABA*ABC*/ABD*/ABE*/ABF
PORT1=AB0*/AB1*/AB2*AB3*AB4*AB5*AB6*/AB7*AB8*AB9*
    ABA*ABB*ABC*/ABD*/ABE*/ABF
```

TL/L/9991-18

```
MEMORY

             11 1111 1111 2222 2222 2233
   0123 4567 8901 2345 6789 0123 4567 8901

BEG*FPLT PAL16L2   8


 24 -X-X -X-X X--X X--X X-X- X--- -XX- X-X- /AB0*/AB1*/AB2*AB3*AB4*AB5*AB6*
 25 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 26 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 27 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 28 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 29 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 30 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 32 -XX- -X-X X--X X--X X-X- X-X- -XX- X-X- AB0*/AB1*/AB2*AB3*AB4*AB5*AB6*/
 33 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 34 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 35 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 36 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 37 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 38 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX


END*FPLT

 LEGEND:  X : FUSE NOT BLOWN (L,N,0)    - : FUSE BLOWN    (H,P,1)
          0 : PHANTOM FUSE   (L,N,0)    0 : PHANTOM FUSE  (H,P,1)

 NUMBER OF FUSES BLOWN =    32
```

TL/L/9991-21

7.3 Memory-Mapped I/O Address Decoder (Continued)


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
1 AB$_0$
2 AB$_1$
19 ABF
3 AB$_2$
18 ABE
4 AB$_3$
17 ABD
24 25 26 27 28 29 30 31
16 PORT 0
5 AB$_4$
32 33 34 35 36 37 38 39
15 PORT 1
6 AB$_5$
14 ABC
7 AB$_6$
13 ABB
8 AB$_7$
12 ABA
9 AB$_8$
11 AB$_9$

TL/L/9991-22

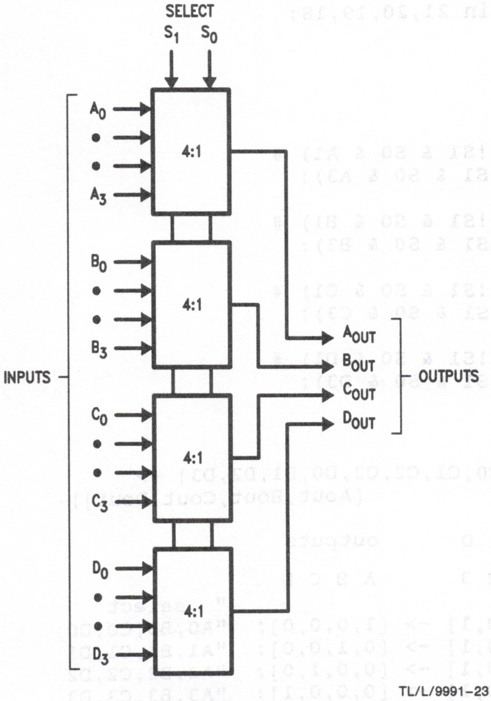
FIGURE 7.3.2. PAL16L2 Logic Diagram Showing Address Decoder Fuse Pattern

7.4 Quad 4-to-1 Multiplexer

DESCRIPTION

Widely used in computer and data communications circuits, multiplexers route one of several input banks to an output, based on the condition of select inputs. This particular version has 4 input banks, each 4-bits wide *(Figure 7.4.1)*; therefore, two select lines are required to choose 1 of 4 inputs, as shown in the function table of *Figure 7.4.2*. Possible applications for our multiplexer include bus selection in a multibus computer environment, or data manipulation in an arithmetic/logic circuit.

With a total of 16 multiplexer inputs and two Select inputs, this design is well suited for the GAL20V8. The pinout chosen for this example is shown in *Figure 7.4.3*; actual pin placement of the multiplexer outputs is not critical since the versatility of the GAL20V8 allows the designer to choose that combination of output pins that best suits the board layout. The device was programmed using ABEL; the logic design input files are shown in *Figure 7.4.4*, with reduced equations shown in the document-generator file of *Figure 7.4.5*. The 'fuse' map is shown in *Figure 7.4.6*.

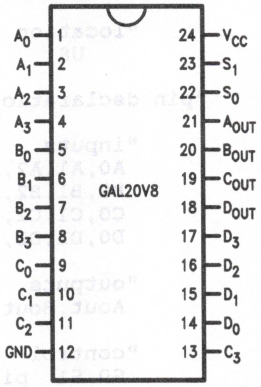

SELECT
S_1
S_0
A_0
A_3
B_0
B_3
C_0
C_3
D_0
D_3
4:1
4:1
4:1
4:1
INPUTS
A_{OUT}
B_{OUT}
C_{OUT}
D_{OUT}
OUTPUTS

TL/L/9991-23

FIGURE 7.4.1. Block Diagram

S_1	S_0	A_{OUT}	B_{OUT}	C_{OUT}	D_{OUT}
0	0	A_0	B_0	C_0	D_0
0	1	A_1	B_1	C_1	D_1
1	0	A_2	B_2	C_2	D_2
1	1	A_3	B_3	C_3	D_3

FIGURE 7.4.2. Function Table


A_0 1
A_1 2
A_2 3
A_3 4
B_0 5
B_1 6
B_2 7
B_3 8
C_0 9
C_1 10
C_2 11
GND 12
24 V_{CC}
23 S_1
22 S_0
21 A_{OUT}
20 B_{OUT}
19 C_{OUT}
18 D_{OUT}
17 D_3
16 D_2
15 D_1
14 D_0
13 C_3
GAL20V8

TL/L/9991-24

FIGURE 7.4.3. Pinout Diagram

7.4 Quad 4-to-1 Multiplexer (Continued)

```
module   quad_4to1_mux

title   'ABEL INPUT FILE
         Quad 4 to 1 Multiplexer in a GAL20V8       April 17, 1986
         National Semiconductor                           Joe Eng'

        "device declaration

              "location        keyword        device code
                 U8            device          'P20V8S';

        "pin declaration

              "inputs
               A0,A1,A2,A3    pin 1,2,3,4;
               B0,B1,B2,B3    pin 5,6,7,8;
               C0,C1,C2,C3    pin 9,10,11,13;
               D0,D1,D2,D3    pin 14,15,16,17;

              "outputs
               Aout,Bout,Cout,Dout  pin 21,20,19,18;

              "control
               S0,S1  pin 22,23;
equations

         Aout = (!S1 & !S0 & A0) # (!S1 & S0 & A1) #
                (S1 & !S0 & A2) # (S1 & S0 & A3);

         Bout = (!S1 & !S0 & B0) # (!S1 & S0 & B1) #
                (S1 & !S0 & B2) # (S1 & S0 & B3);

         Cout = (!S1 & !S0 & C0) # (!S1 & S0 & C1) #
                (S1 & !S0 & C2) # (S1 & S0 & C3);

         Dout = (!S1 & !S0 & D0) # (!S1 & S0 & D1) #
                (S1 & !S0 & D2) # (S1 & S0 & D3);

test_vectors

  ([S1,S0,A0,A1,A2,A3,B0,B1,B2,B3,C0,C1,C2,C3,D0,D1,D2,D3] ->
                                              [Aout,Bout,Cout,Dout])

" S S A     A B     B C     C D     D      outputs
" 1 0 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3      A B C D
                                                         "  select
 [0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1] -> [1,0,0,0];  "A0,B0,C0,D0
 [0,1,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1] -> [0,1,0,0];  "A1,B1,C1,D1
 [1,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1] -> [0,0,1,0];  "A2,B2,C2,D2
 [1,1,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1] -> [0,0,0,1];  "A3,B3,C3,D3

 [0,0,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1] -> [1,1,1,0];  "A0,B0,C0,D0
 [0,1,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1] -> [1,1,0,1];  "A1,B1,C1,D1
 [1,0,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1] -> [1,0,1,1];  "A2,B2,C2,D2
 [1,1,1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,1] -> [0,1,1,1];  "A3,B3,C3,D3
```

TL/L/9991-25

FIGURE 7.4.4. ABEL Input File

```
ABEL(tm) Version 1.19  -  Document Generator
Quad 4 to 1 Multiplexer in a GAL20V8          April 17, 1986
National Semiconductor                              Joe Eng
Equations for Module quad_4to1_mux

Device U8


         Reduced Equations:

            Aout = (A0 & !S0 & !S1
                 # A1 & S0 & !S1
                 # A2 & !S0 & S1
                 # A3 & S0 & S1);


            Bout = (B0 & !S0 & !S1
                 # B1 & S0 & !S1
                 # B2 & !S0 & S1
                 # B3 & S0 & S1);


            Cout = (C0 & !S0 & !S1
                 # C1 & S0 & !S1
                 # C2 & !S0 & S1
                 # C3 & S0 & S1);


            Dout = (D0 & !S0 & !S1
                 # D1 & S0 & !S1
                 # D2 & !S0 & S1
                 # D3 & S0 & S1);
```

TL/L/9991-26

FIGURE 7.4.5. Reduced ABEL Equations

7.4 Quad 4-to-1 Multiplexer (Continued)

```
QP24* QF2706*
L0000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
1101111011101111111111111111111111111111
0111111011011111111111111111111111111111
1111010111101111111111111111111111111111
1111110101011111111111111111111111111111
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
1111111011100111111111111111111111111111
1111111011011111011111111111111111111111
1111110111101111111101111111111111111111
1111110111011111111111110111111111111111
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
1111111011101111111111111111011111111111
1111111011011111111111111111111101111111
1111110111101111111111111111111111110111
1111110111011111111111111111111111111101
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
1111111011101111111111111111111111011111
1111111011011111111111111111111011111111
1111110111101111111111111101111111111111
1111110111011111111111011111111111111111
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000*
L2560
01111000*
L2568
0000000000000000000000000000000000000000000000000000000000000000*
L2632
1*
L2633
0*
L2634
0*
L2635
0*
L2636
0*
L2637
1*
L2638
1*
L2639
1*
L2640
1111111111111111111111111111111111111111111111111111111111111111*
L2704
10*
V0001 10000100001N00001LLLH00N*
V0002 10000100001N00001LLHL10N*
V0003 10000100001N00001LHLL01N*
V0004 10000100001N00001HLLL11N*
V0005 11101101101N10111LHHH00N*
V0006 11101101101N10111HLHH10N*
V0007 11101101101N10111HHLH01N*
V0008 11101101101N10111HHHL11N*
C5127*
```

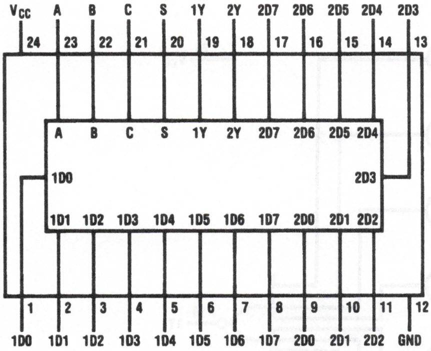
TL/L/9991-27

FIGURE 7.4.6. 'Fuse' Map

7.5 Dual 8-to-1 Multiplexer

The Dual 8:1 Mux selects one of eight inputs, D0 through D7, specified by three binary select inputs, A, B and C. The true data is output on Y when strobed by S. The circuit is implemented using a PAL20L2.

LOGIC SYMBOL



V_{CC}
A
B
C
S
1Y
2Y
2D7
2D6
2D5
2D4
2D3
24
23
22
21
20
19
18
17
16
15
14
13
1D0
2D3
1D1
1D2
1D3
1D4
1D5
1D6
1D7
2D0
2D1
2D2
1
2
3
4
5
6
7
8
9
10
11
12
GND

TL/L/9991-28

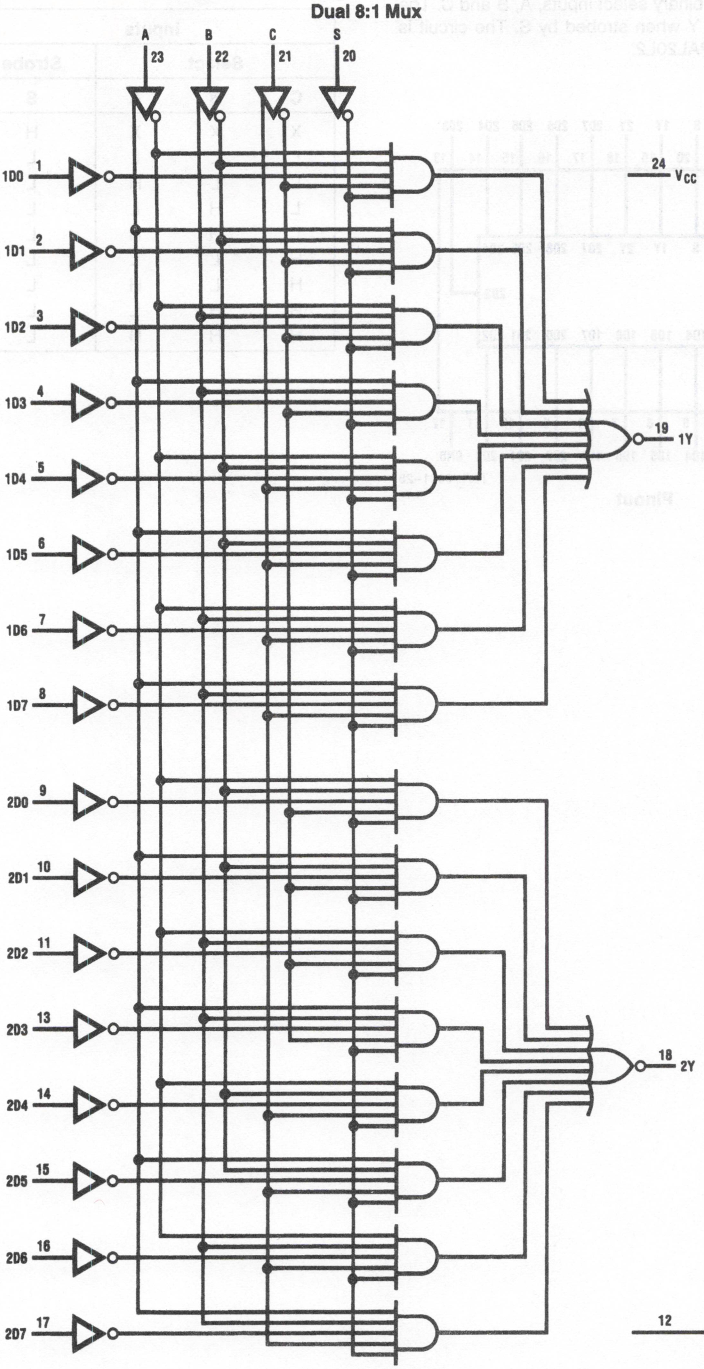
Pinout

FUNCTION TABLE

Inputs				Output
Select			Strobe	Y
C	B	A	S	
X	X	X	H	H
L	L	L	L	D0
L	L	H	L	D1
L	H	L	L	D2
L	H	H	L	D3
H	L	L	L	D4
H	L	H	L	D5
H	H	L	L	D6
H	H	H	L	D7

7.5 Dual 8-to-1 Multiplexer (Continued)

LOGIC DIAGRAM


Dual 8:1 Mux
A
23
B
22
C
21
S
20
1D0
1
1D1
2
1D2
3
1D3
4
1D4
5
1D5
6
1D6
7
1D7
8
2D0
9
2D1
10
2D2
11
2D3
13
2D4
14
2D5
15
2D6
16
2D7
17
24
V_CC
19
1Y
18
2Y
12

TL/L/9991-29

7.5 Dual 8-to-1 Multiplexer (Continued)

PLAN™ INPUT FILE

```
PAL20L2; DUAL 8-TO-1 MULTIPLEXER

1D0 1D1 1D2 1D3 1D4 1D5 1D6 1D7 2D0 2D1 2D2 GND
2D3 2D4 2D5 2D6 2D7 2Y  1Y  S   C   B   A   VCC

/1Y = /1D0*/C*/B*/A*/S
    + /1D1*/C*/B* A*/S
    + /1D2*/C* B*/A*/S
    + /1D3*/C* B* A*/S
    + /1D4* C*/B*/A*/S
    + /1D5* C*/B* A*/S
    + /1D6* C* B*/A*/S
    + /1D7* C* B* A*/S

/2Y = /2D0*/C*/B*/A*/S
    + /2D1*/C*/B* A*/S
    + /2D2*/C* B*/A*/S
    + /2D3*/C* B* A*/S
    + /2D4* C*/B*/A*/S
    + /2D5* C*/B* A*/S
    + /2D6* C* B*/A*/S
    + /2D7* C* B* A*/S
```

TL/L/9991-F0

PLAN™ JEDEC FILE

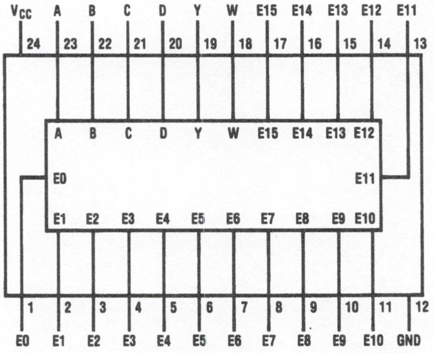
```
PLAN v3.12   09-08-1988    16:59
Source filename: C:MUX8T01.BEQ    Device: PAL20L2
 DUAL 8-TO-1 MULTIPLEXER *
QP24* QF640* F0*
L000
1110111011101110111011111111111111111111
1011110111101110111011111111111111111111
1111101011011110111011111111111111111111
1111110110011110111011111111111111111111
1111111011101001111011111111111111111111
1111110111101101101011111111111111111111
1111111011011101111010111111111111111111
1111110111011101111011111011111111111111
1111111011101110111011111111101111111111
1111110111101110111011111111111110111111
1111111011011110111011111111111111111011
1111110111011110111011111111111111111110
1111111011101101111011111111111111101111
1111110111101101111011111111111101111111
1111111011011101111011111110111111111111
1111110111011101111011101111111111111111*
C408E* 9DBE
```

TL/L/9991-F1

7.6 16-to-1 Multiplexer

The 16:1 Mux selects one of sixteen inputs, E0 through E15, specified by four binary select inputs, A, B, C and D. The true data is output on Y and the inverted data on W. The circuit is implemented using a PAL20C1.

LOGIC SYMBOL


V_{CC}
A
B
C
D
Y
W
E15
E14
E13
E12
E11
24
23
22
21
20
19
18
17
16
15
14
13
A
B
C
D
Y
W
E15
E14
E13
E12
E0
E11
E1
E2
E3
E4
E5
E6
E7
E8
E9
E10
1
2
3
4
5
6
7
8
9
10
11
12
E0
E1
E2
E3
E4
E5
E6
E7
E8
E9
E10
GND

TL/L/9991-30

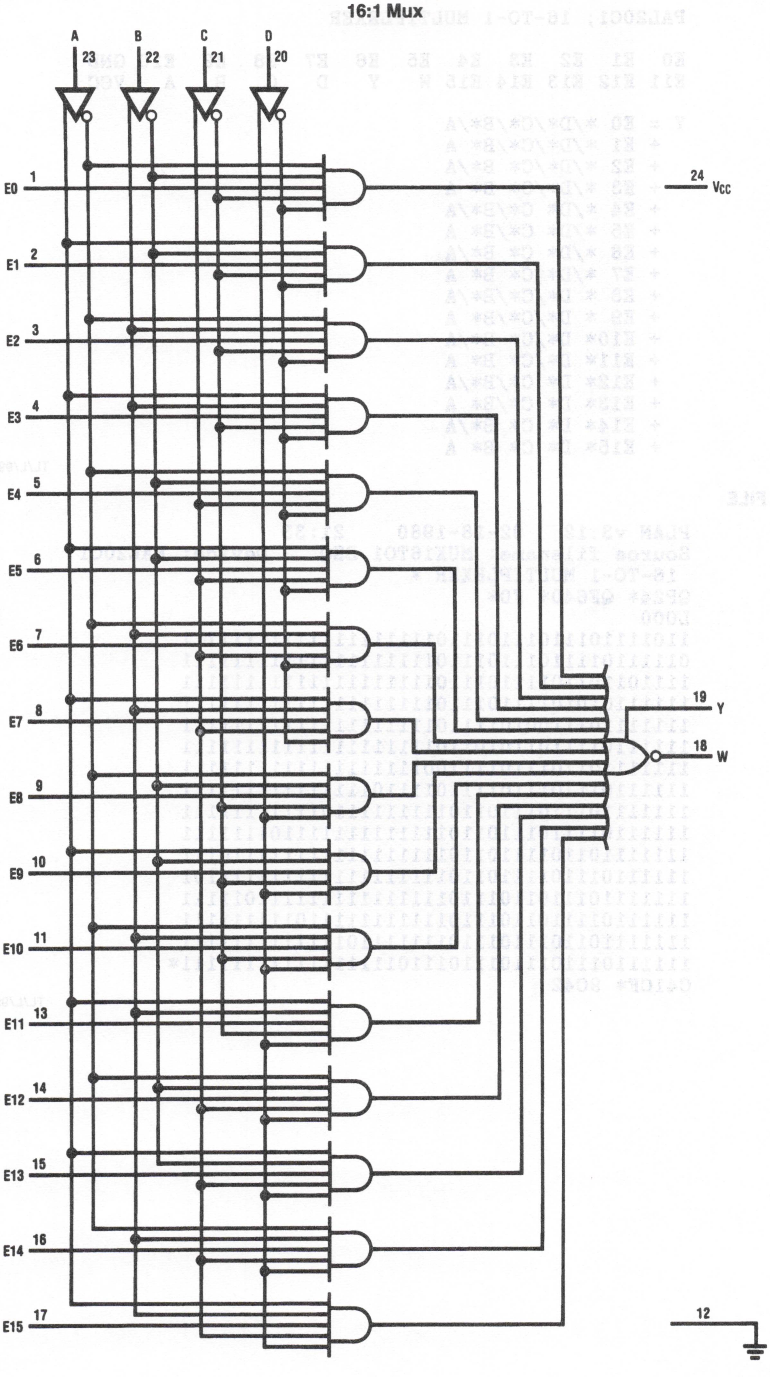
Pinout

FUNCTION TABLE

Input Select				Output	
D	C	B	A	W	Y
L	L	L	L	$\overline{E0}$	E0
L	L	L	H	$\overline{E1}$	E1
L	L	H	L	$\overline{E2}$	E2
L	L	H	H	$\overline{E3}$	E3
L	H	L	L	$\overline{E4}$	E4
L	H	L	H	$\overline{E5}$	E5
L	H	H	L	$\overline{E6}$	E6
L	H	H	H	$\overline{E7}$	E7
H	L	L	L	$\overline{E8}$	E8
H	L	L	H	$\overline{E9}$	E9
H	L	H	L	$\overline{E10}$	E10
H	L	H	H	$\overline{E11}$	E11
H	H	L	L	$\overline{E12}$	E12
H	H	L	H	$\overline{E13}$	E13
H	H	H	L	$\overline{E14}$	E14
H	H	H	H	$\overline{E15}$	E15

7.6 16-to-1 Multiplexer (Continued)

LOGIC DIAGRAM


16:1 Mux
A
B
C
D
23
22
21
20
E0 1
E1 2
E2 3
E3 4
E4 5
E5 6
E6 7
E7 8
E8 9
E9 10
E10 11
E11 13
E12 14
E13 15
E14 16
E15 17
24 Vcc
19 Y
18 W
12

TL/L/9991-31

7.6 16-to-1 Multiplexer (Continued)

PLAN™ INPUT FILE

```
PAL20C1; 16-TO-1 MULTIPLEXER

E0  E1  E2  E3  E4  E5  E6  E7  E8  E9  E10 GND
E11 E12 E13 E14 E15 W   Y   D   C   B   A   VCC

Y = E0 */D*/C*/B*/A
  + E1 */D*/C*/B* A
  + E2 */D*/C* B*/A
  + E3 */D*/C* B* A
  + E4 */D* C*/B*/A
  + E5 */D* C*/B* A
  + E6 */D* C* B*/A
  + E7 */D* C* B* A
  + E8 * D*/C*/B*/A
  + E9 * D*/C*/B* A
  + E10* D*/C* B*/A
  + E11* D*/C* B* A
  + E12* D* C*/B*/A
  + E13* D* C*/B* A
  + E14* D* C* B*/A
  + E15* D* C* B* A
```

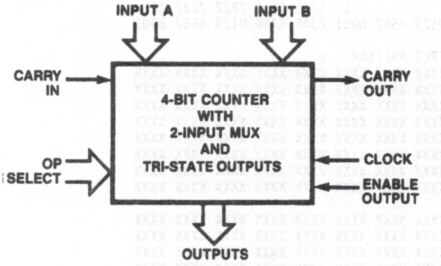
TL/L/9991-F2

PLAN™ JEDEC FILE

```
PLAN v3.12   02-18-1980    21:33
Source filename: MUX16TO1.BEQ    Device: PAL20C1
 16-TO-1 MULTIPLEXER *
QP24* QF640* F0*
L000
1101111011101110111011111111111111111111
0111110111101110111011111111111111111111
1111011011011110111011111111111111111111
1111110101011110111011111111111111111111
1111111011100101111011111111111111111111
1111101111011010110111111111111111111111
1111111011011101111001111111111111111111
1111101110111011101111011111111111111111
1111111011101110110111111111101111111111
1111101111011101101111111111111101111111
1111111011011110110111111111111111110111
1111101110111101101111111111111111111101
1111111011101101110111111111111111011111
1111101111011011101111111111111011111111
1111111011011101110111111101111111111111
1111101110111011101110111111111111111111*
C41CF* 9C42
```

TL/L/9991-F3

7.7 4-Bit Counter with 2 Parallel Load Ports


INPUT A
INPUT B
CARRY IN
CARRY OUT
4-BIT COUNTER WITH 2-INPUT MUX AND TRI-STATE OUTPUTS
OP SELECT
CLOCK
ENABLE OUTPUT
OUTPUTS

TL/L/9991-32

FIGURE 7.7.1. Four-Bit Counter with Two-Input Multiplexer

```
PALASM VERSION 1.5

PAL16R4
PAT0034
4 BIT COUNTER WITH2 INPUT MUX
NSC
CLOCK A0 A1 A2 A3 B0 B1 B2 B3 GND
/E COUT I1 Q3 Q2 Q1 Q0 I0 CIN VCC
/Q0:=/I1*/I0*/Q0 + /I1*I0*/A0 + I1*/I0*/B0 +
I1*I0*/CIN*/Q0 + I1*I0*CIN*Q0
/Q1:=/I1*/I0*/Q1 + /I1*I0*/A1 + I1*/I0*/B1 +
I1*I0*/CIN*/Q1 + I1*I0*CIN*Q1*Q0 + I1*I0*/Q1*/Q0
/Q2:=/I1*/I0*/Q2 + /I1*I0*/A2 + I1*/I0*B2 + I1*I0*/CIN*/Q2 +
I1*I0*CIN*Q2*Q1*Q0 + I1*I0*/Q2*/Q1 + I1*I0*/Q2*/Q0
/Q3:=/I1*/I0*/Q3 + /I1*I0*/A3 + I1*/I0*/B3 + I1*I0*/CIN*/Q3 +
I1*I0*CIN*Q3*Q2*Q1*Q0 + I1*I0*/Q3*/Q2 + I1*I0*/Q3*/Q1 +
I1*I0*/Q3*/Q0
IF(VCC)/COUT = /CIN + /Q3 + /Q2 + /Q1 + /Q0
```

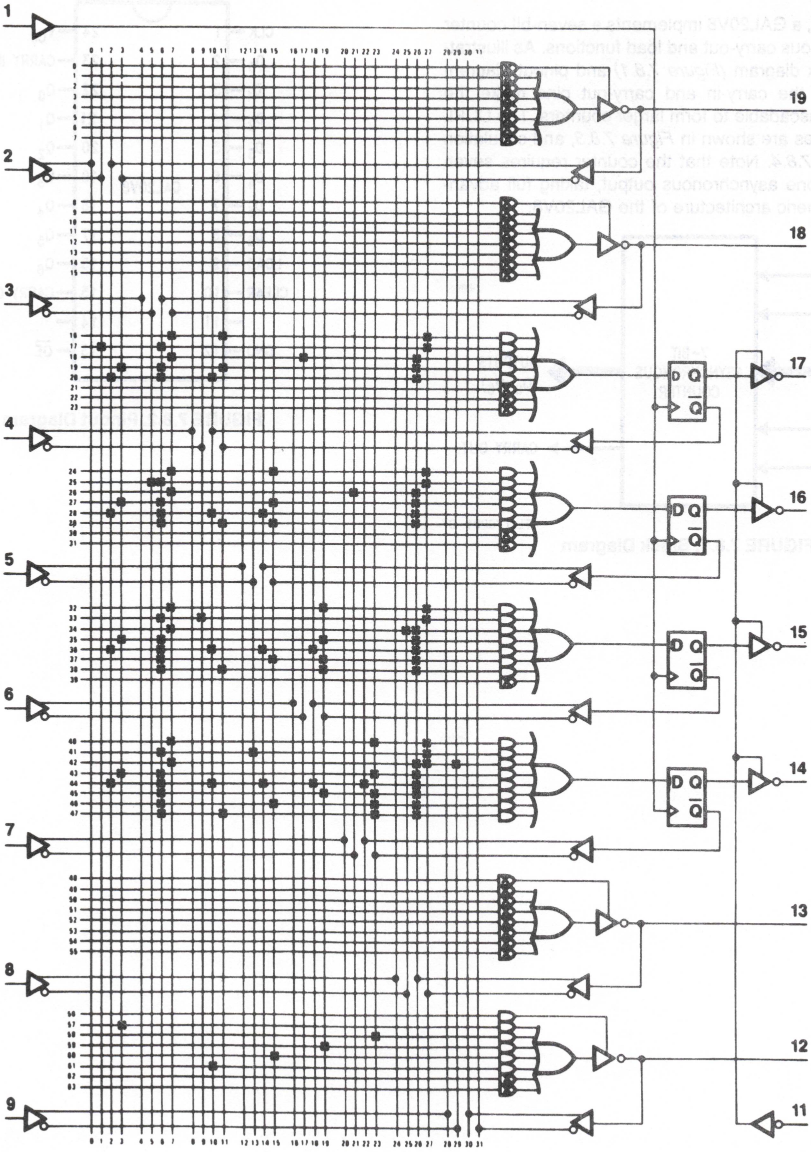
TL/L/9991-70

7.7 4-Bit Counter with 2 Parallel Load Ports (Continued)

```
 4 BIT COUNTER WITH2 INPUT MUX

                11 1111 1111 2222 2222 2233
    0123 4567 8901 2345 6789 0123 4567 8901

BEG*FPLT PAL16R4   8
  0 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  1 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  2 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  3 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  4 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  5 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  6 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  7 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

  8 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
  9 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 10 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 11 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 12 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 13 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 14 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 15 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 16 ---- ---X ---X ---- ---- ---- ---X ---- /I1*/IO*/QO
 17 -X-- --X- ---- ---- ---- ---- ---X ---- /I1*IO*/AO
 18 ---- ---X ---- ---- -X-- ---- --X- ---- I1*/IO*/BO
 19 ---X --X- ---X ---- ---- ---- --X- ---- I1*IO*/CIN*/QO
 20 --X- --X- --X- ---- ---- ---- --X- ---- I1*IO*CIN*QO
 21 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 22 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 23 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 24 ---- ---X ---- ---X ---- ---- ---X ---- /I1*/IO*/Q1
 25 ---- -XX- ---- ---- ---- ---- ---X ---- /I1*IO*/A1
 26 ---- ---X ---- ---- ---- -X-- --X- ---- I1*/IO*/B1
 27 ---X --X- ---- ---X ---- ---- --X- ---- I1*IO*/CIN*/Q1
 28 --X- --X- --X- --X- ---- ---- --X- ---- I1*IO*CIN*Q1*QO
 29 ---- --X- ---X ---X ---- ---- --X- ---- I1*IO*/Q1*/QO
 30 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 32 ---- ---X ---- ---- ---X ---- ---X ---- /I1*/IO*/Q2
 33 ---- --X- -X-- ---- ---- ---- ---X ---- /I1*IO*/A2
 34 ---- ---X ---- ---- ---- ---- X-X- ---- I1*/IO*B2
 35 ---X --X- ---- ---- ---X ---- --X- ---- I1*IO*/CIN*/Q2
 36 --X- --X- --X- --X- --X- ---- --X- ---- I1*IO*CIN*Q2*Q1*QO
 37 ---- --X- ---- ---X ---X ---- --X- ---- I1*IO*/Q2*/Q1
 38 ---- --X- ---X ---- ---X ---- --X- ---- I1*IO*/Q2*/QO
 39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 40 ---- ---X ---- ---- ---- ---X ---X ---- /I1*/IO*/Q3
 41 ---- --X- ---- -X-- ---- ---- ---X ---- /I1*IO*/A3
 42 ---- ---X ---- ---- ---- ---- --X- -X-- I1*/IO*/B3
 43 ---X --X- ---- ---- ---- ---X --X- ---- I1*IO*/CIN*/Q3
 44 --X- --X- --X- --X- --X- --X- --X- ---- I1*IO*CIN*Q3*Q2*Q1*QO
 45 ---- --X- ---- ---- ---X ---X --X- ---- I1*IO*/Q3*/Q2
 46 ---- --X- ---- ---X ---- ---X --X- ---- I1*IO*/Q3*/Q1
 47 ---- --X- ---X ---- ---- ---X --X- ---- I1*IO*/Q3*/QO

 48 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 49 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 50 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 51 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 52 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 53 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 54 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 55 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 56 ---- ---- ---- ---- ---- ---- ---- ----
 57 ---X ---- ---- ---- ---- ---- ---- ---- /CIN
 58 ---- ---- ---- ---- ---- ---X ---- ---- /Q3
 59 ---- ---- ---- ---- ---X ---- ---- ---- /Q2
 60 ---- ---- ---- ---X ---- ---- ---- ---- /Q1
 61 ---- ---- ---X ---- ---- ---- ---- ---- /QO
 62 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 63 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

END*FPLT

 LEGEND:  X : FUSE NOT BLOWN (L,N,0)   - : FUSE BLOWN   (H,P,1)

 NUMBER OF FUSES BLOWN =  921
```

TL/L/9991-35

7.7 4-Bit Counter with 2 Parallel Load Ports (Continued)


1
2
3
4
5
6
7
8
9
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
19
18
17
16
15
14
13
12
11
D Q
Q̄

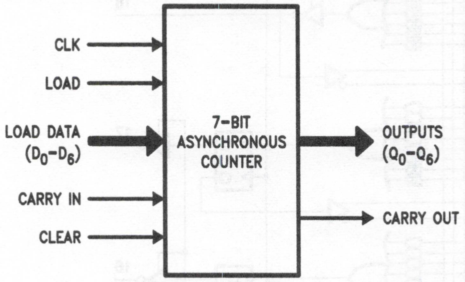
TL/L/9991-36

FIGURE 7.7.2. PAL16R4 Logic Diagram Showing Counter Pattern

7.8 7-Bit Counter with Parallel Load

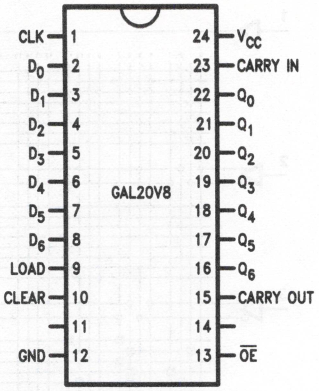
DESCRIPTION

In this example, a GAL20V8 implements a seven-bit counter with asynchronous carry-out and load functions. As illustrated in the block diagram *(Figure 7.8.1)* and pinout diagram *(Figure 7.8.2)*, the carry-in and carry-out pins make the counter fully cascadable to form larger counters. The CUPL design input files are shown in *Figure 7.8.3*, and simulation files in *Figure 7.8.4*. Note that the counter requires seven registers and one asynchronous output, taking full advantage of the generic architecture of the GAL20V8.


CLK
LOAD
LOAD DATA
(D_0-D_6)
CARRY IN
CLEAR
7-BIT
ASYNCHRONOUS
COUNTER
OUTPUTS
(Q_0-Q_6)
CARRY OUT

TL/L/9991-37

FIGURE 7.8.1. Block Diagram


CLK 1
D_0 2
D_1 3
D_2 4
D_3 5
D_4 6
D_5 7
D_6 8
LOAD 9
CLEAR 10
11
GND 12
GAL20V8
24 V_{CC}
23 CARRY IN
22 Q_0
21 Q_1
20 Q_2
19 Q_3
18 Q_4
17 Q_5
16 Q_6
15 CARRY OUT
14
13 $\overline{OE}$

TL/L/9991-38

FIGURE 7.8.2. Pinout Diagram

```
/*******************************************************************/
/*                                                                 */
/*                        CUPL INPUT FILE                          */
/*                DESIGN INPUT FOR 7-BIT COUNTER                   */
/*                                                                 */
/*******************************************************************/
/*        ALLOWABLE TARGET DEVICE:   GAL20V8                       */
/*******************************************************************/
PARTNO      7BITCNT ;
NAME        7-BIT COUNTER ;
REV         01 ;
DATE        10/08/85 ;
DESIGNER    Joe Engineer;
COMPANY     National Semiconductor;
ASSEMBLY    3A-27 ;
LOCATION    U06 ;

PIN 1 = CLK ;              /* CLOCK INPUT */
PIN 2 = D0 ;               /* DATA0 INPUT */
PIN 3 = D1 ;               /* DATA1 INPUT */
PIN 4 = D2 ;               /* DATA2 INPUT */
PIN 5 = D3 ;               /* DATA3 INPUT */
PIN 6 = D4 ;               /* DATA4 INPUT */
PIN 7 = D5 ;               /* DATA5 INPUT */
PIN 8 = D6 ;               /* DATA6 INPUT */
PIN 9 = LD ;               /* LOAD CONTROL */
PIN 10 = CLEAR;            /* ASYNCHRONOUS CARRY-IN */

PIN 13 = !OE ;             /* OUTPUT ENABLE */
PIN 15 = CARRYOUT ;
PIN 16 = Q6 ;              /* COUNTER MSB */
PIN 17 = Q5;
PIN 18 = Q4 ;
PIN 19 = Q3 ;
PIN 20 = Q2 ;
PIN 21 = Q1 ;
PIN 22 = Q0 ;              /* COUNTER LSB */
PIN 23 = CARRYIN ;         /* CARRY-IN FOR CASCADING */
```

TL/L/9991-39

FIGURE 7.8.3. CUPL Design Input File

7.8 7-Bit Counter with Parallel Load (Continued)

```
Q0.D = (LD   & D0                                       /* LOAD D0 */
    # !LD & !Q0 & CARRYIN) & !CLEAR;                    /*  TOGGLE */

Q1.D = (LD   & D1                                       /* LOAD D1 */
    # !LD& !Q1 &   Q0 & CARRYIN                         /*  TOGGLE */
    # !LD&  Q1 & !Q0) & !CLEAR;                         /*   HOLD  */

Q2.D = (LD   & D2                                       /* LOAD D2 */
    # !LD& !Q2 &   Q1 & Q0 & CARRYIN                    /* TOGGLE  */
    # !LD&  Q2 & !Q1                                    /*  HOLD   */
    # !LD&  Q2 & !Q0) & !CLEAR;                         /*  HOLD   */

Q3.D = (LD   & D3                                       /* LOAD D3 */
    # !LD& !Q3 &   Q2 & Q1 & Q0 & CARRYIN               /*  TOGGLE */
    # !LD&  Q3 & !Q2                                    /*   HOLD  */
    # !LD&  Q3 & !Q1                                    /*   HOLD  */
    # !LD&  Q3 & !Q0) & !CLEAR;                         /*   HOLD  */

Q4.D = (LD   & D4                                       /* LOAD D4 */
    # !LD& !Q4& Q3 & Q2 & Q1 & Q0 & CARRYIN             /*  TOGGLE */
    # !LD& Q4 & !Q3                                     /*   HOLD  */
    # !LD& Q4 & !Q2                                     /*   HOLD  */
    # !LD& Q4 & !Q1                                     /*   HOLD  */
    # !LD& Q4 & !Q0) & !CLEAR;                          /*   HOLD  */

Q5.D = (LD   & D5                                       /* LOAD D5 */
    # !LD& !Q5& Q4 & Q3 & Q2 & Q1 & Q0
      & CARRYIN                                         /*  TOGGLE */
    # !LD& Q5 & !Q4                                     /*   HOLD  */
    # !LD& Q5 & !Q3                                     /*   HOLD  */
    # !LD& Q5 & !Q2                                     /*   HOLD  */
    # !LD& Q5 & !Q1                                     /*   HOLD  */
    # !LD& Q5 & !Q0) & !CLEAR;                          /*   HOLD  */

Q6.D = (LD   & D6                                       /* LOAD D6 */
    # !LD& !Q6& Q5 & Q4 & Q3 & Q2 & Q1 & Q0
      & CARRYIN                                         /*  TOGGLE */
    # !LD& Q6 & !Q5                                     /*   HOLD  */
    # !LD& Q6 & !Q4                                     /*   HOLD  */
    # !LD& Q6 & !Q3                                     /*   HOLD  */
    # !LD& Q6 & !Q2                                     /*   HOLD  */
    # !LD& Q6 & !Q1                                     /*   HOLD  */
    # !LD& Q6 & !Q0) & !CLEAR;                          /*   HOLD  */


CARRYOUT = !LD & Q6 & Q5 & Q4 & Q3 & Q2 & Q1 & Q0
           & CARRYIN;                                   /* CARRY-OUT */
```

TL/L/9991-40

FIGURE 7.8.3. CUPL Design Input File (Continued)

7.8 7-Bit Counter with Parallel Load (Continued)

```
/**************************************************************/
/*                                                            */
/*                      CUPL INPUT FILE                       */
/*               SIMULATION FOR 7-BIT COUNTER                 */
/*                                                            */
/**************************************************************/
/*            ALLOWABLE TARGET DEVICE: GAL20V8                */
/**************************************************************/
PARTNO       7BITCNT ;
NAME         7-BIT COUNTER ;
REV          01 ;
DATE         10/08/85 ;
DESIGNER     Joe Engineer;
COMPANY      National Semiconductor;
ASSEMBLY     3A-27 ;
LOCATION     U06 ;

ORDER:

  CLK, !OE, CLEAR, LD, CARRYIN, D6, D5, D4, D3, D2, D1, D0, Q6,
  Q5, Q4, Q3, Q2, Q1, Q0, CARRYOUT;
VECTORS:

$msg"                           C ";
$msg" C ! C   C                 O ";
$msg" L O L L I DDDDDDD QQQQQQQ U ";
$msg" K E R D N 6543210 6543210 T ";
$msg" --------------------------- ";
      0 1 X X X XXXXXXX ZZZZZZZ X  /* TEST HI-Z  */
      C 0 1 X X XXXXXXX LLLLLLL L  /* TEST CLEAR */
      C 0 0 1 X 1111111 HHHHHHH L  /* LOAD ONES  */
      C 0 0 1 X 0000000 LLLLLLL L  /* LOAD ZEROS */
      C 0 0 0 1 XXXXXXX LLLLLLH L  /* COUNT=1 */
      C 0 0 0 1 XXXXXXX LLLLLHL L  /* COUNT=2 */
      C 0 0 0 1 XXXXXXX LLLLLHH L  /* COUNT=3 */
      C 0 0 0 1 XXXXXXX LLLLHLL L  /* COUNT=4 */
      C 0 0 0 1 XXXXXXX LLLLHLH L  /* COUNT=5 */
      C 0 0 0 1 XXXXXXX LLLLHHL L  /* COUNT=6 */
      C 0 0 0 1 XXXXXXX LLLLHHH L  /* COUNT=7 */
      C 0 0 0 1 XXXXXXX LLLHLLL L  /* COUNT=8 */
      C 0 0 0 1 XXXXXXX LLLHLLH L  /* COUNT=9 */
      C 0 0 0 1 XXXXXXX LLLHLHL L  /* COUNT=10 */
      C 0 0 0 1 XXXXXXX LLLHLHH L  /* COUNT=11 */
      C 0 0 0 1 XXXXXXX LLLHHLL L  /* COUNT=12 */
      C 0 0 0 1 XXXXXXX LLLHHLH L  /* COUNT=13 */
      C 0 0 0 1 XXXXXXX LLLHHHL L  /* COUNT=14 */
      C 0 0 0 1 XXXXXXX LLLHHHH L  /* COUNT=15 */
      C 0 0 0 1 XXXXXXX LLHLLLL L  /* COUNT=16 */
      C 0 0 0 1 XXXXXXX LLHLLLH L  /* COUNT=17 */
      C 0 0 0 1 XXXXXXX LLHLLHL L  /* COUNT=18 */
      C 0 0 0 1 XXXXXXX LLHLLHH L  /* COUNT=19 */
      C 0 0 0 1 XXXXXXX LLHLHLL L  /* COUNT=20 */
      C 0 0 0 1 XXXXXXX LLHLHLH L  /* COUNT=21 */
      C 0 0 0 1 XXXXXXX LLHLHHL L  /* COUNT=22 */
      C 0 0 0 1 XXXXXXX LLHLHHH L  /* COUNT=23 */
      C 0 0 0 1 XXXXXXX LLHHLLL L  /* COUNT=24 */
      C 0 0 0 1 XXXXXXX LLHHLLH L  /* COUNT=25 */
      C 0 0 0 1 XXXXXXX LLHHLHL L  /* COUNT=26 */
      C 0 0 0 1 XXXXXXX LLHHLHH L  /* COUNT=27 */
      C 0 0 0 1 XXXXXXX LLHHHLL L  /* COUNT=28 */
      C 0 0 0 1 XXXXXXX LLHHHLH L  /* COUNT=29 */
      C 0 0 0 1 XXXXXXX LLHHHHL L  /* COUNT=30 */
      C 0 0 0 1 XXXXXXX LLHHHHH L  /* COUNT=31 */
      C 0 0 0 1 XXXXXXX LHLLLLL L  /* COUNT=32 */
      C 0 0 0 1 XXXXXXX LHLLLLH L  /* COUNT=33 */
      C 0 0 1 X 0111111 LHHHHHH L  /* LOAD=63 TO OBSERVE MSB TQGGLE */
      C 0 0 0 1 XXXXXXX HLLLLLL L  /* COUNT=64, OBSERVE MSB */
      C 0 0 0 1 XXXXXXX HLLLLLH L  /* COUNT=65, OBSERVE MSB */
      C 0 0 1 X 1111110 HHHHHHL L  /* LOAD=126 TO OBSERVE CARRY */
      C 0 0 0 1 XXXXXXX HHHHHHH H  /* COUNT=127, OBSERVE CARRY  */
      C 0 0 0 1 XXXXXXX LLLLLLL L  /* COUNT=0, OBSERVE CARRY    */
```

TL/L/9991-50

FIGURE 7.8.4. CUPL Simulation File

7.9 8-Bit Counter with Parallel Load

The 8-bit synchronous counter is used in many systems. The input A0 serves as a mode control with LOW for LOAD operation and HIGH for count operation. Input A1 enables the LOAD operation when A0 is set in the LOAD mode and doesn't care when the count mode is chosen. This enables the counter to be cascaded as a multibyte counter with the capability of loading individual bytes from a simple byte-wide data bus and a common clock. /CIN is the carry input and /COUT is the carry output.

This function is most efficiently implemented using a PAL20X8.

PLAN™ INPUT FILE

```
PAL20X8; 8-BIT SYNCHRONOUS COUNTER

CLK A0     X0  X1  X2  X3  X4  X5  X6  X7  A1   GND
/EN /COUT Y7  Y6  Y5  Y4  Y3  Y2  Y1  Y0  /CIN VCC

/Y0 := /A1*/A0*/Y0 +  A0*/Y0
   :+:   A1*/A0*/X0 +  A0*/CIN
/Y1 := /A1*/A0*/Y1 +  A0*/Y1
   :+:   A1*/A0*/X1 +  A0*/CIN*/Y0
/Y2 := /A1*/A0*/Y2 +  A0*/Y2
   :+:   A1*/A0*/X2 +  A0*/CIN*/Y0*/Y1
/Y3 := /A1*/A0*/Y3 +  A0*/Y3
   :+:   A1*/A0*/X3 +  A0*/CIN*/Y0*/Y1*/Y2
/Y4 := /A1*/A0*/Y4 +  A0*/Y4
   :+:   A1*/A0*/X4 +  A0*/CIN*/Y0*/Y1*/Y2*/Y3
/Y5 := /A1*/A0*/Y5 +  A0*/Y5
   :+:   A1*/A0*/X5 +  A0*/CIN*/Y0*/Y1*/Y2*/Y3*/Y4
/Y6 := /A1*/A0*/Y6 +  A0*/Y6
   :+:   A1*/A0*/X6 +  A0*/CIN*/Y0*/Y1*/Y2*/Y3*/Y4*/Y5
/Y7 := /A1*/A0*/Y7 +  A0*/Y7
   :+:   A1*/A0*/X7 +  A0*/CIN*/Y0*/Y1*/Y2*/Y3*/Y4*/Y5*/Y6
/COUT =/CIN*/Y0*/Y1*/Y2*/Y3*/Y4*/Y5*/Y6*/Y7
```

TL/L/9991-F4

PLAN™ JEDEC FILE

```
PLAN v3.12   09-08-1988    14:00
Source filename: COUNT8.BEQ    Device: PAL20X8
 8-BIT SYNCHRONOUS COUNTER *
QP24* QF1600* F0*
L0160
1011111011111111111111111111111111111011
0111111011111111111111111111111111111111
1011101111111111111111111111111111110111
0110111111111111111111111111111111111111
1011111111101111111111111111111111111011
0111111111101111111111111111111111111111
1011111110111111111111111111111111110111
0110111011111111111111111111111111111111
1011111111111110111111111111111111111011
0111111111111110111111111111111111111111
1011111111111011111111111111111111110111
0110111011101111111111111111111111111111
1011111111111111111011111111111111111011
0111111111111111111011111111111111111111
1011111111111111101111111111111111110111
0110111011101110111111111111111111111111
1011111111111111111111101111111111111011
0111111111111111111111101111111111111111
1011111111111111111110111111111111110111
0110111011101110111011111111111111111111
1011111111111111111111111110111111111011
0111111111111111111111111110111111111111
1011111111111111111111111011111111110111
0110111011101110111011101111111111111111
1011111111111111111111111111111011111011
0111111111111111111111111111111011111111
1011111111111111111111111111101111110111
0110111011101110111011101110111111111111
1011111111111111111111111111111111101011
0111111111111111111111111111111111101111
1011111111111111111111111111111110110111
0110111011101110111011101110111011111111
1111111111111111111111111111111111111111
1110111011101110111011101110111011101111*
C9816* 298A
```

TL/L/9991-F5

7.10 10-Bit Up/Down Counter

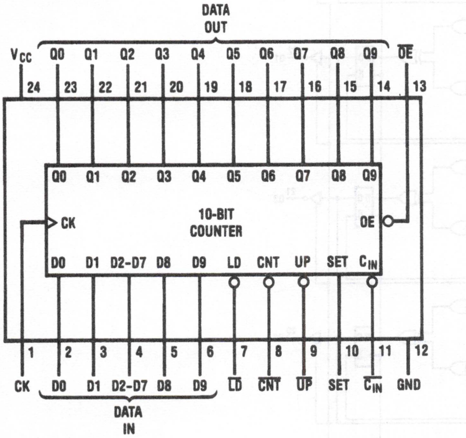
The ten-bit counter can count up, count down, set, and load 2 LSB's, 2 MSB's and 6 middle bits high or low as a group. All operations are synchronous with the clock. SET overrides LOAD, COUNT and HOLD. LOAD overrides COUNT. COUNT is conditional on C_{IN}, otherwise it holds.

All outputs are enabled when $\overline{OE}$ is low, otherwise HIGH-Z. This circuit is implemented using a PAL20X10.

FUNCTION TABLE

$\overline{OE}$	CK	SET	LD	$\overline{CNT}$	$\overline{C_{IN}}$	$\overline{UP}$	D9–D0	Q9–Q0	Operation
H	X	X	X	X	X	X	X	Z	Hi-Z
L	↑	H	X	X	X	X	X	H	Set all HIGH
L	↑	L	L	X	X	X	D	D	LOAD D
L	↑	L	H	H	X	X	X	Q	HOLD
L	↑	L	H	L	H	X	X	Q	HOLD
L	↑	L	H	L	L	L	X	Q plus 1	Count UP
L	↑	L	H	L	L	H	X	Q minus 1	Count DN

LOGIC SYMBOL


DATA OUT
V_{CC}
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9
$\overline{OE}$
24 23 22 21 20 19 18 17 16 15 14 13
CK
10-BIT COUNTER
OE
D0 D1 D2-D7 D8 D9 LD CNT UP SET C_{IN}
1 2 3 4 5 6 7 8 9 10 11 12
CK D0 D1 D2-D7 D8 D9 $\overline{LD}$ $\overline{CNT}$ $\overline{UP}$ SET $\overline{C_{IN}}$ GND
DATA IN

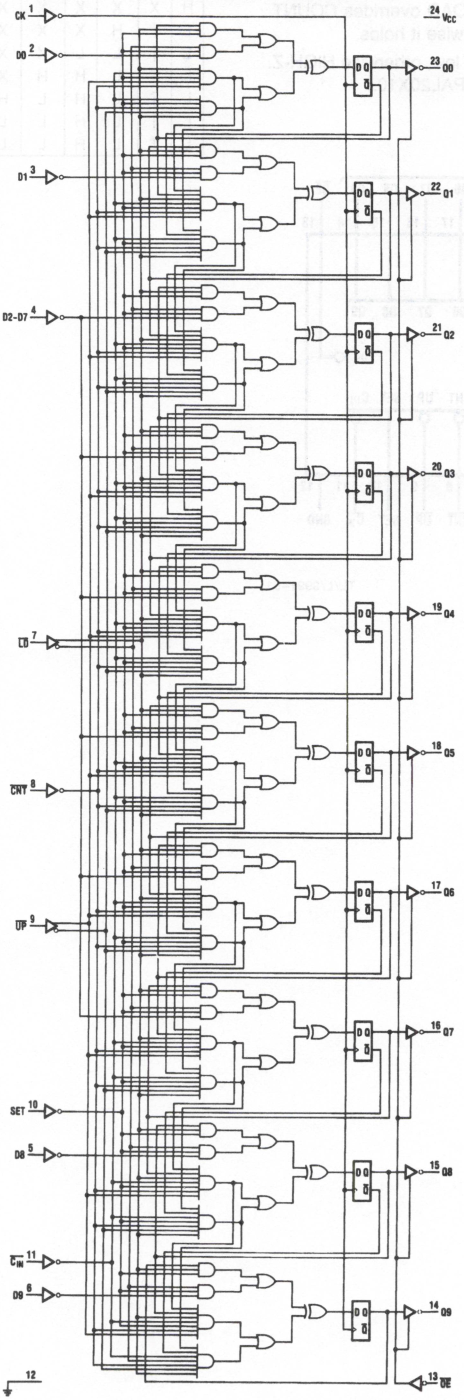
TL/L/9991-51

Pinout

7.10 10-Bit Up/Down Counter (Continued)

LOGIC DIAGRAM

10-Bit Up/Down Counter


CK 1
D0 2
D1 3
D2-D7 4
$\overline{LD}$ 7
$\overline{CNT}$ 8
$\overline{UP}$ 9
SET 10
D8 5
$\overline{C_{IN}}$ 11
D9 6
12
24 V_{CC}
23 Q0
22 Q1
21 Q2
20 Q3
19 Q4
18 Q5
17 Q6
16 Q7
15 Q8
14 Q9
13 $\overline{OE}$
D Q
$\overline{Q}$

TL/L/9991-52

7.10 10-Bit Up/Down Counter (Continued)

PLAN™ INPUT FILE

```
PAL20X10; 10-BIT SYNCHRONOUS UP/DOWN COUNTER

CLK  D0   D1   D2D7 /LD  /CNT /UP  SET  D8   /CIN D9   GND
/OE  Q9   Q8   Q7   Q6   Q5   Q4   Q3   Q2   Q1   Q0   VCC

/Q0 :=  /Q0*LD*/SET + /LD*/SET*/D0
    :+: LD*/CIN*/SET*/CNT* UP
      + LD*/CIN*/SET*/CNT*/UP

/Q1 :=  /Q1*LD*/SET + /LD*/SET*/D1
    :+: LD*/CIN*/SET*/CNT* UP*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q0

/Q2 :=  /Q2*LD*/SET + /LD*/SET*/D2D7
    :+: LD*/CIN*/SET*/CNT* UP*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q1* Q0

/Q3 :=  /Q3*LD*/SET + /LD*/SET*/D2D7
    :+: LD*/CIN*/SET*/CNT* UP*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q2* Q1* Q0

/Q4 :=  /Q4*LD*/SET + /LD*/SET*/D2D7
    :+: LD*/CIN*/SET*/CNT* UP*/Q3*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q3* Q2* Q1* Q0

/Q5 :=  /Q5*LD*/SET + /LD*/SET*/D2D7
    :+: LD*/CIN*/SET*/CNT* UP*/Q4*/Q3*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q4* Q3* Q2* Q1* Q0

/Q6 :=  /Q6*LD*/SET + /LD*/SET*/D2D7
    :+: LD*/CIN*/SET*/CNT* UP*/Q5*/Q4*/Q3*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q5* Q4* Q3* Q2* Q1* Q0

/Q7 :=  /Q7*LD*/SET + /LD*/SET*/D2D7
    :+: LD*/CIN*/SET*/CNT* UP*/Q6*/Q5*/Q4*/Q3*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q6* Q5* Q4* Q3* Q2* Q1* Q0

/Q8 :=  /Q8*LD*/SET + /LD*/SET*/D8
    :+: LD*/CIN*/SET*/CNT* UP*/Q7*/Q6*/Q5*/Q4*/Q3*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q7* Q6* Q5* Q4* Q3* Q2* Q1* Q0

/Q9 :=  /Q9*LD*/SET + /LD*/SET*/D9
    :+: LD*/CIN*/SET*/CNT* UP*/Q8*/Q7*/Q6*/Q5*/Q4*/Q3*/Q2*/Q1*/Q0
      + LD*/CIN*/SET*/CNT*/UP* Q8* Q7* Q6* Q5* Q4* Q3* Q2* Q1* Q0
```

TL/L/9991-F6

7.10 10-Bit Up/Down Counter (Continued)

PLAN™ JEDEC FILE

```
PLAN v3.12   02-18-1980    21:42
Source filename: COUNT10.BEQ    Device: PAL20X10
 10-BIT SYNCHRONOUS UP/DOWN COUNTER *
QP24* QF1600* F0*
L0000
1110111111110111111111111011111111111111
1011111111111011111111111011111111111111
1111111111110111101110111011111110111111
1111111111110111101101111011111110111111
1111111011110111111111111011111111111111
1111101111111011111111111011111111111111
1110111111110111101110111011111110111111
1101111111110111101101111011111110111111
1111111111100111111111111011111111111111
1111111110111011111111111011111111111111
1110111011110111101110111011111110111111
1101110111110111101101111011111110111111
1111111111110110111111111011111111111111
1111111110111011111111111011111111111111
1110111011100111101110111011111110111111
1101110111010111101101111011111110111111
1111111111110111111011111011111111111111
1111111110111011111111111011111111111111
1110111011100110101110111011111110111111
1101110111010101101101111011111110111111
1111111111110111111111101011111111111111
1111111110111011111111111011111111111111
1110111011100110101010111011111110111111
1101110111010101100101111011111110111111
1111111111110111111111111010111111111111
1111111110111011111111111011111111111111
1110111011100110101010101011111110111111
1101110111010101100101011011111110111111
1111111111110111111111111011111011111111
1111111110111011111111111011111111111111
1110111011100110101010101010111110111111
1101110111010101100101011001111110111111
1111111111110111111111111011111111101111
1111111111111011111111111011101111111111
1110111011100110101010101010111010111111
1101110111010101100101011001110110111111
1111111111110111111111111011111111111110
1111111111111011111111111011111111111011
1110111011100110101010101010111010101111
1101110111010101100101011001110110011111*
CAE56* 5A2E
```

TL/L/9991-F7

7.11 8-Bit Cascadable Shift Register

This is an 8-bit synchronous shift register with parallel load and hold capability. Two function select inputs (I_0, I_1) provide one of four operations which occur synchronously on the rising edge of the clock (CK).

The LOAD operation loads the input (D_7–D_0) into the output register (Q_7–Q_0). The HOLD operation holds the previous value regardless of clock transitions. The SHIFT LEFT operation shifts the output register, Q, one bit to the left; Q_0 is replaced by LIRO. RILO outputs Q_7. The SHIFT RIGHT operation shifts the output register, Q, one bit to the right; Q_7 is replaced by RILO. LIRO outputs Q_0.

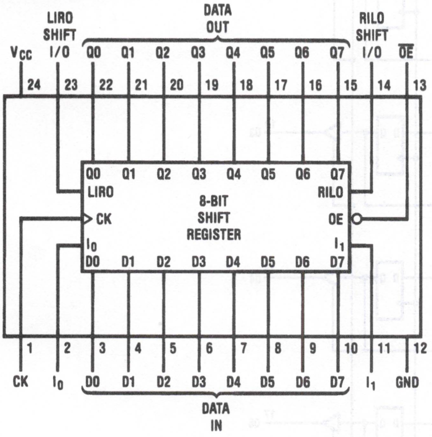
The output register (Q_7–Q_0) is enabled when $\overline{OE}$ is LOW, and disabled (HI-Z) when $\overline{OE}$ is HIGH. Two or more octal shift registers may be cascaded to provide larger shift registers.

This function is identical to that of the 6-bit cascadable shift register example described throughout Section 4. The 8-bit shifter, however, requires a PAL20X8. Notice that because I_0 and I_1 select only one product-term at a time in each output function, the XOR gates may be treated as ordinary OR gates to generate 4-term functions.

FUNCTION TABLE

$\overline{OE}$	CK	I_1	I_0	D_7–D_0	Q_7–Q_0	Operation
H	X	X	X	X	Z	HI-Z
L	↑	L	L	X	L	HOLD
L	↑	L	H	X	SR(Q)	SHIFT RIGHT
L	↑	H	L	X	SL(Q)	SHIFT LEFT
L	↑	H	H	D	D	LOAD

LOGIC SYMBOL


LIRO SHIFT I/O
DATA OUT
RILO SHIFT I/O
V_{CC}
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
$\overline{OE}$
24 23 22 21 20 19 18 17 16 15 14 13
LIRO
RILO
CK
OE
8-BIT SHIFT REGISTER
I_0
I_1
D0 D1 D2 D3 D4 D5 D6 D7
1 2 3 4 5 6 7 8 9 10 11 12
CK I_0 D0 D1 D2 D3 D4 D5 D6 D7 I_1 GND
DATA IN

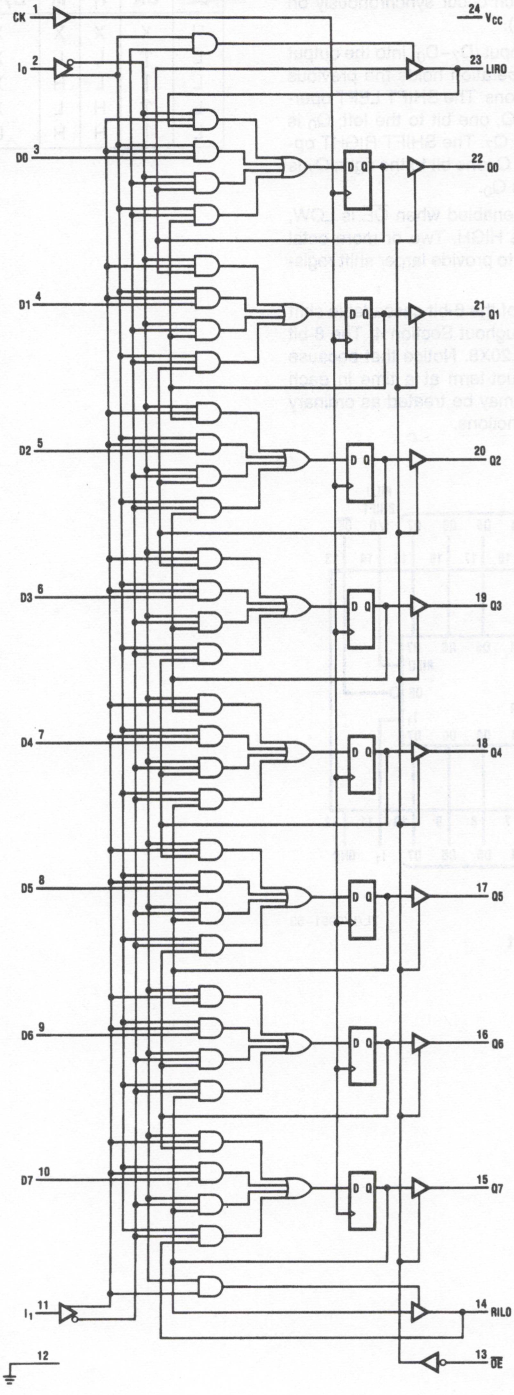
TL/L/9991-53

Pinout

7.11 8-Bit Cascadable Shift Register (Continued)

LOGIC DIAGRAM

Octal Shift Register


CK 1
I0 2
D0 3
D1 4
D2 5
D3 6
D4 7
D5 8
D6 9
D7 10
I1 11
12
24 VCC
23 LIRO
22 Q0
21 Q1
20 Q2
19 Q3
18 Q4
17 Q5
16 Q6
15 Q7
14 RILO
13 OE
D Q

TL/L/9991-54

7.11 8-Bit Cascadable Shift Register (Continued)

PLAN™ INPUT FILE

```
PAL20X8; 8-BIT CASCADABLE SHIFT REGISTER

CLK I0   D0   D1   D2   D3   D4   D5   D6   D7   I1   GND
/G  RILO Q7   Q6   Q5   Q4   Q3   Q2   Q1   Q0   LIRO VCC

;NOTE: I0 = SHIFT RIGHT; I1 = SHIFT LEFT

/Q0 := /Q0*/I0*/I1 + /Q1*I0*/I1 :+: /LIRO*/I0*I1 + /D0*I0*I1
/Q1 := /Q1*/I0*/I1 + /Q2*I0*/I1 :+: /Q0*/I0*I1 + /D1*I0*I1
/Q2 := /Q2*/I0*/I1 + /Q3*I0*/I1 :+: /Q1*/I0*I1 + /D2*I0*I1
/Q3 := /Q3*/I0*/I1 + /Q4*I0*/I1 :+: /Q2*/I0*I1 + /D3*I0*I1
/Q4 := /Q4*/I0*/I1 + /Q5*I0*/I1 :+: /Q3*/I0*I1 + /D4*I0*I1
/Q5 := /Q5*/I0*/I1 + /Q6*I0*/I1 :+: /Q4*/I0*I1 + /D5*I0*I1
/Q6 := /Q6*/I0*/I1 + /Q7*I0*/I1 :+: /Q5*/I0*I1 + /D6*I0*I1
/Q7 := /Q7*/I0*/I1 + /RILO*I0*/I1 :+: /Q6*/I0*I1 + /D7*I0*I1
/LIRO = /Q0
LIRO.TRST = I0*/I1
/RILO = /Q5
RILO.TRST = /I0*I1
```

TL/L/9991-F8

PLAN™ JEDEC FILE

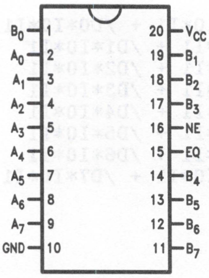
```
PLAN v3.12    09-08-1988     13:53
Source filename: SHIFT8.BEQ     Device: PAL20X8
 8-BIT CASCADABLE SHIFT REGISTER *
QP24* QF1600* F0*
L0000
0111111111111111111111111111111111111011
1111110111111111111111111111111111111111*
L0160
1011111011111111111111111111111111111011
0111111111101111111111111111111111111011
1010111111111111111111111111111111110111
0111101111111111111111111111111111110111
1011111111101111111111111111111111111011
0111111111111110111111111111111111111011
1011111011111111111111111111111111110111
0111111110111111111111111111111111110111
1011111111111110111111111111111111111011
0111111111111111111011111111111111111011
1011111111101111111111111111111111110111
0111111111111011111111111111111111110111
1011111111111111111011111111111111111011
0111111111111111111111101111111111111011
1011111111111110111111111111111111110111
0111111111111111101111111111111111110111
1011111111111111111111101111111111111011
0111111111111111111111111110111111111011
1011111111111111111011111111111111110111
0111111111111111111110111111111111110111
1011111111111111111111111110111111111011
0111111111111111111111111111111011111011
1011111111111111111111101111111111110111
0111111111111111111111111011111111110111
1011111111111111111111111111111011111011
0111111111111111111111111111111111101011
1011111111111111111111111110111111110111
0111111111111111111111111111101111110111
1011111111111111111111111111111111101011
0111111111111111111111111111111111111010
1011111111111111111111111111111011110111
0111111111111111111111111111111111110111
1011111111111111111111111111111111110111
1111111111111111111111111101111111111111*
CA879* 3BBC
```

TL/L/9991-F9

7.12 8-Bit Equality Comparator

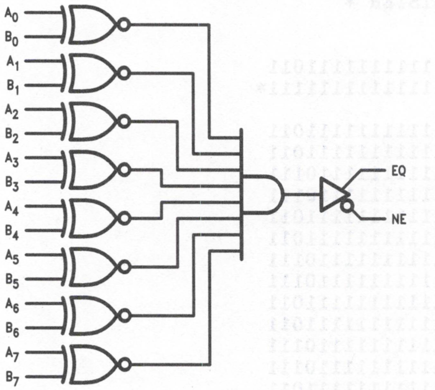
The equality comparator takes two 8-bit data words as inputs and produces one combinatorial output (EQ) which becomes active when the two input words match. Due to the large number of product terms required to implement this function, a PAL16C1 is the appropriate device to use. The complement output (NE) is also provided.

PINOUT



TL/L/9991-G0

LOGIC DIAGRAM



TL/L/9991-G1

Note: To express this function in the sum-of-products form required by the PAL16C1, it is necessary to apply the identity function

$$X \oplus Y = X\overline{Y} + \overline{X}Y$$

as well as DeMorgan's Theorem

$$\overline{A} \bullet \overline{B} = \overline{A + B}$$

to the above logic definition.

PLAN™ INPUT FILE

```
PAL16C1; 8-BIT EQUALITY COMPARATOR

B0   A0   A1   A2   A3   A4   A5   A6   A7   GND
B7   B6   B5   B4   EQ   NE   B3   B2   B1   VCC

NE   = A0*/B0 + /A0*B0
     + A1*/B1 + /A1*B1
     + A2*/B2 + /A2*B2
     + A3*/B3 + /A3*B3
     + A4*/B4 + /A4*B4
     + A5*/B5 + /A5*B5
     + A6*/B6 + /A6*B6
     + A7*/B7 + /A7*B7
```

TL/L/9991-G2

PLAN™ JEDEC FILE

```
PLAN v3.12   09-08-1988     17:32
Source filename: COMPARE8.BEQ    Device: PAL16C1
 8-BIT EQUALITY COMPARATOR *
QP20* QF512* F0*
L000
01101111111111111111111111111111
10011111111111111111111111111111
11110110111111111111111111111111
11111001111111111111111111111111
11111111011011111111111111111111
11111111100111111111111111111111
11111111111101101111111111111111
11111111111110011111111111111111
11111111111111110110111111111111
11111111111111111001111111111111
11111111111111111111011011111111
11111111111111111111100111111111
11111111111111111111111101101111
11111111111111111111111110011111
11111111111111111111111111110110
11111111111111111111111111111001*
C3BC4* 85E5
```

TL/L/9991-G3

7.13 8-Bit Barrel Shifter

DESCRIPTION

The barrel shifter *(Figure 7.13.1)* is a specialized shift register that rotates data a selectable number of bit positions out of the most-significant bit and back into the least-significant bit—thus the name. Typical applications of a barrel shifter are floating-point arithmetic and display rotation on a graphics terminal.

Since our barrel shifter has 8 data inputs and 8 registered outputs, as well as control signals, the GAL20V8 is the PLD of choice. The shift-select inputs (S_0, S_1, S_2) determine the number of positions shifted, as described in the function table of *Figure 7.13.2*. The block diagram is shown in *Figure 7.13.3*, and the pinout in *Figure 7.13.4*. The clock (CLK) input gates input data synchronously to the output registers, and the output enable (OE) allows TRI-STATE® buffering of the Q outputs. The one remaining input is used for a reset (RS) function.

The ABEL design input files shown in *Figure 7.13.5* may appear tedious, but simply enumerate the eight different bit-shift possibilities for each output.

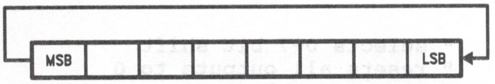

MSB
LSB
TL/L/9991-55

FIGURE 7.13.1. Barrel Shift Rotation

S_2	S_1	S_0	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	1	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_7
0	1	0	D_5	D_4	D_3	D_2	D_1	D_0	D_7	D_6
0	1	1	D_4	D_3	D_2	D_1	D_0	D_7	D_6	D_5
1	0	0	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4
1	0	1	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3
1	1	0	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2
1	1	1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1

FIGURE 7.13.2. Function Table

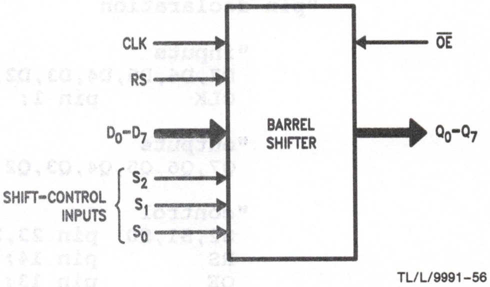

CLK
RS
D0–D7
SHIFT-CONTROL INPUTS
S2
S1
S0
BARREL SHIFTER
OE
Q0–Q7
TL/L/9991-56

FIGURE 7.13.3. Block Diagram

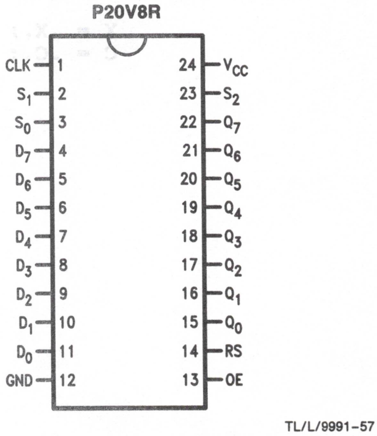

P20V8R
CLK 1
S1 2
S0 3
D7 4
D6 5
D5 6
D4 7
D3 8
D2 9
D1 10
D0 11
GND 12
24 VCC
23 S2
22 Q7
21 Q6
20 Q5
19 Q4
18 Q3
17 Q2
16 Q1
15 Q0
14 RS
13 OE
TL/L/9991-57

FIGURE 7.13.4. Pinout Diagram

7.13 8-Bit Barrel Shifter (Continued)

```
module barrel_shifter_8;

title 'ABEL INPUT FILE
      8-bit Barrel Shifter in a GAL20V8              April 16, 1986
      National Semiconductor                             Joe Eng'

      "device declaration

         "location     keyword    device code
            U9         device      'P20V8R';

      "pin declaration

         "inputs
         D7,D6,D5,D4,D3,D2,D1,D0  pin 4,5,6,7,8,9,10,11;
         CLK         pin 1;

         "outputs
         Q7,Q6,Q5,Q4,Q3,Q2,Q1,Q0  pin 22,21,20,19,18,17,16,15;

         "control
         S2,S1,S0    pin 23,2,3;      " selects 0-7 bit shift
         RS          pin 14;          " resets all outputs to 0
         OE          pin 13;          " output enable

      "constant declaration

         X = .X.;                " simplify 'don't care' constant
         C = .C.;                " simplify 'clock' constant
```

TL/L/9991-45

FIGURE 7.13.5. ABEL Input File

```
equations

        Q0 := !RS & ((!S2 & !S1 & !S0 & D0) #
                     (!S2 & !S1 & S0 & D7) #
                     (!S2 & S1 & !S0 & D6) #
                      (!S2 & S1 & S0 & D5) #
                     (S2 & !S1 & !S0 & D4) #
                      (S2 & !S1 & S0 & D3) #
                      (S2 & S1 & !S0 & D2) #
                       (S2 & S1 & S0 & D1));

        Q1 := !RS & ((!S2 & !S1 & !S0 & D1) #
                     (!S2 & !S1 & S0 & D0) #
                     (!S2 & S1 & !S0 & D7) #
                      (!S2 & S1 & S0 & D6) #
                     (S2 & !S1 & !S0 & D5) #
                      (S2 & !S1 & S0 & D4) #
                      (S2 & S1 & !S0 & D3) #
                       (S2 & S1 & S0 & D2));

        Q2 := !RS & ((!S2 & !S1 & !S0 & D2) #
                     (!S2 & !S1 & S0 & D1) #
                     (!S2 & S1 & !S0 & D0) #
                      (!S2 & S1 & S0 & D7) #
                     (S2 & !S1& !S0 & D6) #
                      (S2 & !S1 & S0 & D5) #
                      (S2 & S1 & !S0 & D4) #
                       (S2 & S1 & S0 & D3));

        Q3 := !RS & ((!S2 & !S1 & !S0 & D3) #
                     (!S2 & !S1 & S0 & D2) #
                     (!S2 & S1 & !S0 & D1)#
                      (!S2 & S1 & S0 & D0)#
                     (S2 & !S1 & !S0 & D7)#
                      (S2 & !S1 & S0& D6)#
                      (S2 & S1 & !S0 & D5)#
                       (S2 & S1 & S0 & D4));

        Q4 := !RS & ((!S2 & !S1 & !S0 & D4) #
                     (!S2 & !S1 & S0 & D3) #
                     (!S2 & S1 & !S0& D2) #
                      (!S2& S1 & S0& D1) #
                     (S2 & !S1 & !S0 & D0) #
                      (S2 & !S1 & S0 &D7)#
                      (S2 & S1 & !S0 & D6) #
                       (S2 & S1 & S0 & D5));
```

TL/L/9991-46

FIGURE 7.13.5. ABEL Input File (Continued)

7.13 8-Bit Barrel Shifter (Continued)

```
        Q5 := !RS & ((!S2 & !S1 & !S0 & D5) #
                      (!S2 & !S1 & S0 & D4) #
                      (!S2 & S1 & !S0 & D3) #
                       (!S2 & S1 & S0 & D2) #
                      (S2 & !S1 & !S0 & D1) #
                       (S2 & !S1 & S0 & D0) #
                       (S2 & S1 & !S0 & D7) #
                        (S2 & S1 & S0 & D6));

        Q6 := !RS & ((!S2 & !S1 & !S0 & D6) #
                      (!S2 & !S1 & S0 & D5) #
                      (!S2 & S1 & !S0 & D4) #
                       (!S2 & S1 & S0 & D3) #
                      (S2 & !S1 & !S0 & D2) #
                       (S2 & !S1 & S0 & D1) #
                       (S2 & S1 & !S0 & D0) #
                        (S2 & S1 & S0 & D7));

        Q7 := !RS & ((!S2 & !S1 & !S0 & D7) #
                      (!S2 & !S1 & S0 & D6) #
                      (!S2 & S1 & !S0 & D5) #
                       (!S2 & S1 & S0 & D4) #
                      (S2 & !S1 & !S0 & D3) #
                       (S2 & !S1 & S0 & D2) #
                       (S2 & S1 & !S0 & D1) #
                        (S2 & S1 & S0 & D0));

test_vectors  ([CLK,OE,RS,S2,S1,S0,D7..D0] -> [Q7..Q0])

   "  C
   "  L O R S   S D             D       Q             Q
   "  K E S 2 1 0 7 6 5 4 3 2 1 0       7 6 5 4 3 2 1 0
     [C,0,1,X,X,X,X,X,X,X,X,X,X,X] -> [0,0,0,0,0,0,0,0];  " set
     [C,0,0,0,0,0,0,0,0,0,1,1,1,1] -> [0,0,0,0,1,1,1,1];  " no shift
     [C,0,0,0,0,1,1,1,1,1,0,0,0,0] -> [1,1,1,0,0,0,0,1];  " shift 1
     [C,0,0,0,1,0,0,0,0,0,1,1,1,1] -> [0,0,1,1,1,1,0,0];  "       2
     [C,0,0,0,1,1,1,1,1,1,0,0,0,0] -> [1,0,0,0,0,1,1,1];  "       3
     [C,0,0,1,0,0,0,0,0,0,1,1,1,1] -> [1,1,1,1,0,0,0,0];  "       4
     [C,0,0,1,0,1,1,1,1,1,0,0,0,0] -> [0,0,0,1,1,1,1,0];  "       5
     [C,0,0,1,1,0,0,0,0,0,1,1,1,1] -> [1,1,0,0,0,0,1,1];  "       6
     [C,0,0,1,1,1,1,1,1,1,0,0,0,0] -> [0,1,1,1,1,0,0,0];  '       7

end barrel_shifter_8
```

TL/L/9991-47

FIGURE 7.13.5. ABEL Input File (Continued)

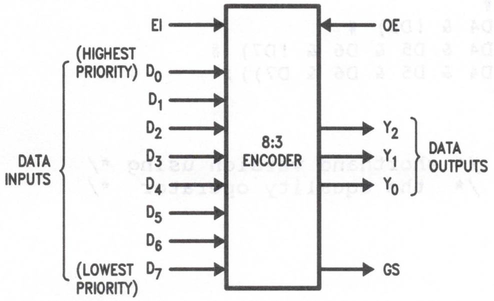
7.14 8-Input Priority Encoder

DESCRIPTION

Encoding is the process of converting several lines of input data into a compact code. This code is then either transmitted along a system bus (usually of limited capacity) to be decoded elsewhere, or stored in memory for future use. A priority encoder functions as a normal encoder when only one input is active; that input's code appears at the output. When more than one input is active, the priority encoder selects the input with the highest priority, ignoring all others. The priority structure is set by the logic equations programmed into the GAL16V8, and the user then makes the appropriate pin assignments.

Our priority encoder *(Figure 7.14.1)* has a few extra features to enhance its capabilities. Both an input enable and an output enable are included to allow for independent control by the input and output circuits. Additionally, a group strobe (GS) output signal is provided to notify the output circuit when a valid output code is present. The truth table for the encoder is shown in *Figure 7.14.2*, and GAL16V8 pinout diagram in *Figure 7.14.3*.

The logic equations for this circuit are quite long. Fortunately, the CUPL programming language, which was used for this example, allows shortcuts in the logic description. As shown in the design input file *(Figure 7.14.4)*, the equation for Y_0 has been written in long form. Y_1 and Y_2 were considerably easier to write, thanks to the use of the 'equality operator.' The simulation file is shown in *Figure 7.14.5*, and CUPL provides the expanded product terms in *Figure 7.14.6*.

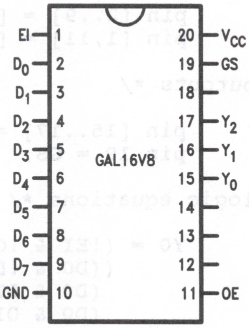


TL/L/9991-58

FIGURE 7.14.1. Block Diagram

EI	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	OE	Y_2	Y_1	Y_0	GS
1	X	X	X	X	X	X	X	X	X	0	0	0	1
X	X	X	X	X	X	X	X	X	1	0	0	0	1
0	0	X	X	X	X	X	X	X	0	0	0	0	0
0	1	0	X	X	X	X	X	X	0	0	0	1	0
0	1	1	0	X	X	X	X	X	0	0	1	0	0
0	1	1	1	0	X	X	X	X	0	0	1	1	0
0	1	1	1	1	0	X	X	X	0	1	0	0	0
0	1	1	1	1	1	0	X	X	0	1	0	1	0
0	1	1	1	1	1	1	0	X	0	1	1	0	0
0	1	1	1	1	1	1	1	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	0	1	1	1	1

FIGURE 7.14.2. Function Table



TL/L/9991-59

FIGURE 7.14.3. Pinout Diagram

7.14 8-Input Priority Encoder (Continued)

```
/*************************************************************/
/*                                                           */
/*                    CUPL INPUT FILE                        */
/*        GAL16V8 - 8 Line to 3 Line Priority Encoder        */
/*                                                           */
/*************************************************************/
        PARTNO          803PLD;
        NAME            8_3ENCOD;
        REV             1;
        DATE            4/18/86;
        DESIGNER        Joe Engineer;
        COMPANY         National Semiconductor;
        ASSEMBLY        Interface Board;
        LOCATION        U09;

/* inputs */

        pin [2..9] = [D0..7];
        pin [1,11] = [EI,OE];     /* Enable Input,Output Enable */

/* outputs */

        pin [15..17] = [Y0..2];
        pin 19 = GS;              /* Group Strobe */

/* logic equations */

        Y0 = (!EI & !OE) &
             ((D0 & !D1) #
              (D0 & D1 & D2 & !D3) #
              (D0 & D1 & D2 & D3 & D4 & !D5) #
              (D0 & D1 & D2 & D3 & D4 & D5 & D6 & !D7) #
              (D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7));

        Y1 = (!EI & !OE) &
             ((D0 & D1 & !D2) #
              [D0..2]:& & !D3 #      /* shorthand version using */
              [D0..5]:& & !D6 #      /*  the equality operator  */
              [D0..6]:& & !D7 #
              [D0..7]:&);

        Y2 = (!EI & !OE) &
             ([D0..3]:& & !D4 #
              [D0..4]:& & !D5 #
              [D0..5]:& & !D6 #
              [D0..6]:& & !D7 #
              [D0..7]:&);

        GS = EI # OE # [D0..7]:&;
```

TL/L/9991-60

FIGURE 7.14.4. Design Input File

7.14 8-Input Priority Encoder (Continued)

```
/*******************************************************************/
/*                                                                 */
/*                      CUPL SIMULATION FILE                       */
/*      GAL16V8 - 8 Line to 3 Line Priority Encoder                */
/*                                                                 */
/*******************************************************************/
          PARTNO              803PLD;
          NAME                8_3ENCOD;
          REV                 1;
          DATE                4/18/86;
          DESIGNER            Joe Engineer;
          COMPANY             National Semiconductor;
          ASSEMBLY            Interface Board;
          LOCATION            U09;

order:
          EI,%1,OE,%2,D0,%1,D1,%1,D2,%1,D3,%1,D4,%1,D5,%1,
                              D6,%1,D7,%2,Y2,%1,Y1,%1,Y0,%2,GS;

vectors:
          /*        D               D  Y   Y      */
          /* EI OE  0 1 2 3 4 5 6 7  2 1 0 GS */

               1 X  X X X X X X X X  L L L  H  /* test Enable Input */
               X 1  X X X X X X X X  L L L  H  /* test Output Enable */
               0 0  1 1 1 1 1 1 1 1  H H H  H  /* all inputs high */

               0 0  0 0 0 0 0 0 0 0  L L L  L  /*    test     */
               0 0  1 0 0 0 0 0 0 0  L L H  L  /*  priority   */
               0 0  1 1 0 0 0 0 0 0  L H L  L  /*  function   */
               0 0  1 1 1 0 0 0 0 0  L H H  L
               0 0  1 1 1 1 0 0 0 0  H L L  L
               0 0  1 1 1 1 1 0 0 0  H L H  L
               0 0  1 1 1 1 1 1 0 0  H H L  L
               0 0  1 1 1 1 1 1 1 0  H H H  L
```

TL/L/9991-61

FIGURE 7.14.5. Simulation File

7.14 8-Input Priority Encoder (Continued)

```
                          CUPL DOCUMENTATION FILE
*******************************************************************************
                                 8_3ENCOD
*******************************************************************************

CUPL            2.10B1
Device          g16v8s  Library DLIB-d-55-8
Created         Fri Apr 18 15:52:32 1986
Name            8_3ENCOD
Partno          803PLD
Revision        1
Date            4/18/86
Designer        Joe Engineer
Company         National Semiconductor
Assembly        Interface Board
Location        U09

===============================================================================
                          Expanded Product Terms
===============================================================================

GS =>
    EI
  # OE
  # D0 & D1 & D2 & D3 & D4 & D5 & D6 & D7

Y0 =>
    D0 & !D1 & !EI & !OE
  # D0 & D1 & D2 & !D3 & !EI & !OE
  # D0 & D1 & D2 & D3 & D4 & !D5 & !EI & !OE
  # D0 & D1 & D2 & D3 & D4 & D5 & D6 & !EI & !OE

Y1 =>
    D0 & D1 & !D2 & !EI & !OE
  # D0 & D1 & D2 & !D3 & !EI & !OE
  # D0 & D1 & D2 & D3 & D4 & D5 & !EI & !OE

Y2 =>
    D0 & D1 & D2 & D3 & !EI & !OE
```

TL/L/9991-62

FIGURE 7.14.6. Expanded Product Terms

7.15 16-Input Priority Encoder

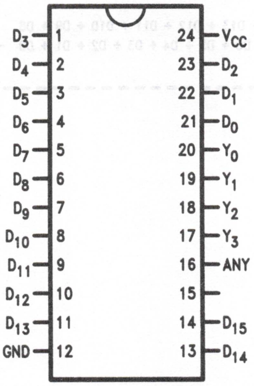
This example describes a priority encoder similar to that in Section 7.14, but with 16 inputs being encoded onto four outputs. Encoding 16 inputs requires 8 product terms on at least 3 of the combinatorial outputs, which is normally not available in any standard TTL PAL device. The solution employs a GAL20V8 used in "Small PAL" mode. But unlike any TTL Small PAL devices, 8 product terms are available on all outputs. Programmable polarity is also very useful in this application to realize a 16-input OR function on the "ANY" output.

Incidentally, unlike the 8-input encoder described in Section 7.14, the inputs in this example are active-high, and D_{15} is the highest priority input.

TRUTH TABLE

Inputs																Outputs				
D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**	**D11**	**D12**	**D13**	**D14**	**D15**	**Y3**	**Y2**	**Y1**	**Y0**	**ANY**
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	H	H	H	H	H	H
X	X	X	X	X	X	X	X	X	X	X	X	X	X	H	L	H	H	H	L	H
X	X	X	X	X	X	X	X	X	X	X	X	X	H	L	L	H	H	L	H	H
X	X	X	X	X	X	X	X	X	X	X	X	H	L	L	L	H	H	L	L	H
X	X	X	X	X	X	X	X	X	X	X	H	L	L	L	L	H	L	H	H	H
X	X	X	X	X	X	X	X	X	X	H	L	L	L	L	L	H	L	H	L	H
X	X	X	X	X	X	X	X	X	H	L	L	L	L	L	L	H	L	L	H	H
X	X	X	X	X	X	X	X	H	L	L	L	L	L	L	L	H	L	L	L	H
X	X	X	X	X	X	X	H	L	L	L	L	L	L	L	L	L	H	H	H	H
X	X	X	X	X	X	H	L	L	L	L	L	L	L	L	L	L	H	H	L	H
X	X	X	X	X	H	L	L	L	L	L	L	L	L	L	L	L	H	L	H	H
X	X	X	X	H	L	L	L	L	L	L	L	L	L	L	L	L	H	L	L	H
X	X	X	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H
X	X	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L	H
X	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H
H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

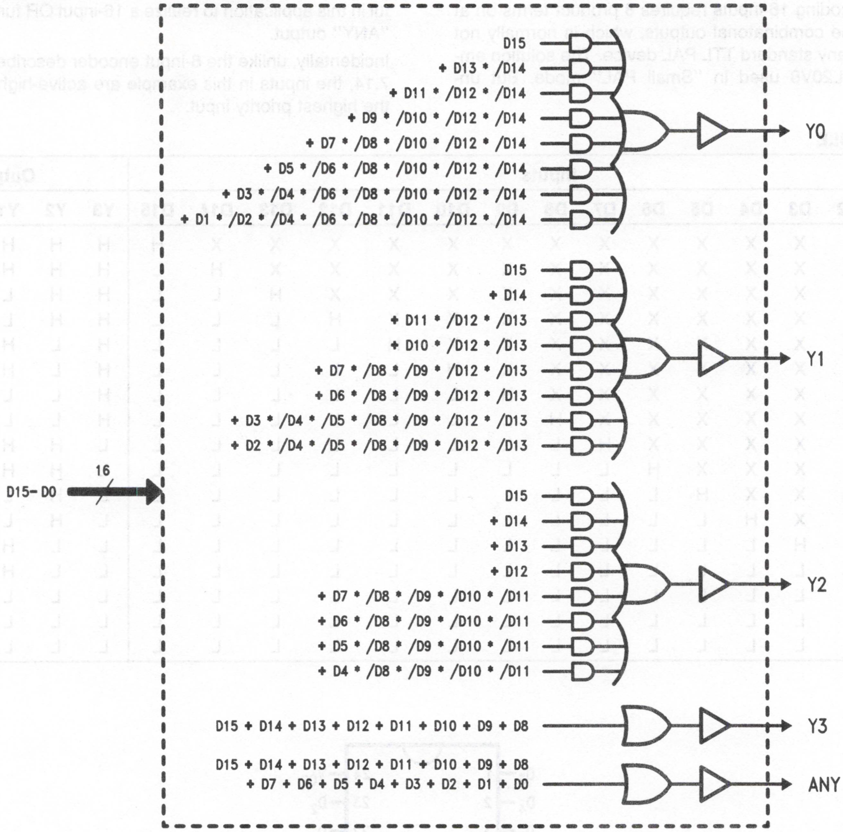
PINOUT



TL/L/9991-G4

7.15 16-Input Priority Encoder (Continued)

LOGIC DEFINITION


D15
+ D13 * /D14
+ D11 * /D12 * /D14
+ D9 * /D10 * /D12 * /D14
+ D7 * /D8 * /D10 * /D12 * /D14
+ D5 * /D6 * /D8 * /D10 * /D12 * /D14
+ D3 * /D4 * /D6 * /D8 * /D10 * /D12 * /D14
+ D1 * /D2 * /D4 * /D6 * /D8 * /D10 * /D12 * /D14
Y0
D15
+ D14
+ D11 * /D12 * /D13
+ D10 * /D12 * /D13
+ D7 * /D8 * /D9 * /D12 * /D13
+ D6 * /D8 * /D9 * /D12 * /D13
+ D3 * /D4 * /D5 * /D8 * /D9 * /D12 * /D13
+ D2 * /D4 * /D5 * /D8 * /D9 * /D12 * /D13
Y1
16
D15- D0
D15
+ D14
+ D13
+ D12
+ D7 * /D8 * /D9 * /D10 * /D11
+ D6 * /D8 * /D9 * /D10 * /D11
+ D5 * /D8 * /D9 * /D10 * /D11
+ D4 * /D8 * /D9 * /D10 * /D11
Y2
D15 + D14 + D13 + D12 + D11 + D10 + D9 + D8
Y3
D15 + D14 + D13 + D12 + D11 + D10 + D9 + D8
+ D7 + D6 + D5 + D4 + D3 + D2 + D1 + D0
ANY

TL/L/9991-63

7.15 16-Input Priority Encoder (Continued)

PLAN™ INPUT FILE

```
GAL20V8; 16-TO-4 PRIORITY ENCODER
D3  D4  D5  D6  D7  D8  D9  D10 D11 D12 D13 GND
D14 D15 NC  ANY Y3  Y2  Y1  Y0  D0  D1  D2  VCC

 Y0 =  D15
    + /D14* D13
    + /D14*/D12* D11
    + /D14*/D12*/D10* D9
    + /D14*/D12*/D10*/D8* D7
    + /D14*/D12*/D10*/D8*/D6* D5
    + /D14*/D12*/D10*/D8*/D6*/D4* D3
    + /D14*/D12*/D10*/D8*/D6*/D4*/D2* D1

 Y1 =  D15
    +  D14
    + /D13*/D12* D11
    + /D13*/D12* D10
    + /D13*/D12*/D9*/D8* D7
    + /D13*/D12*/D9*/D8* D6
    + /D13*/D12*/D9*/D8*/D5*/D4* D3
    + /D13*/D12*/D9*/D8*/D5*/D4* D2

 Y2 =  D15 + D14 + D13 + D12
    + /D11*/D10*/D9*/D8* D7
    + /D11*/D10*/D9*/D8* D6
    + /D11*/D10*/D9*/D8* D5
    + /D11*/D10*/D9*/D8* D4

/Y3 = /D15*/D14*/D13*/D12*/D11*/D10*/D9*/D8

/ANY = /D15*/D14*/D13*/D12*/D11*/D10*/D9*/D8
      */D7*/D6*/D5*/D4*/D3*/D2*/D1*/D0
```

TL/L/9991-G5

7.15 16-Input Priority Encoder (Continued)

PLAN™ JEDEC FILE

```
PLAN v3.12   02-18-1980    22:49
Source filename: ENCODE16.BEQ     Device: GAL20V8
 16-TO-4 PRIORITY ENCODER *
QP24* QF2624* F0*
L0640
1111111111111111111111111111111111011111
1111111111111111111111111111111111110110
1111111111111111111111111111011110111110
1111111111111111111101111011111110111110
1111111111110111101111111011111110111110
1111011110111111101111111011111110111110
1001111110111111101111111011111110111110
1011111010011111101111111011111110111110
1111111111111111111111111111111111011111
1111111111111111111111111111111111111101
1111111111111111111111111111011110111011
1111111111111111111111110111111110111011
1111111111110111101110111111111110111011
1111111011111111101110111111111110111011
1001101111111111101110111111111110111011
1011100111111111101110111111111110111011
1111111111111111111111111111111111011111
1111111111111111111111111111111111111101
1111111111111111111111111111111111101111
1111111111111111111111111111111110111111
1111111111110111101110111011101111111111
1111111011111111101110111011101111111111
1111011111111111101110111011101111111111
0111111111111111101110111011101111111111
1111111111111111101110111011101110101010*
L1920
1010101010101010101110111011101110101010*
L2560
00111000*
L2632
11000001*
L2640
1111111111111111111111111111111111111111
111111111111111111111111*
L2704
10*
C7E99* 02BF
```

TL/L/9991-G6

7.16 Hexadecimal 7-Segment Display Encoder

The increasing use of microcomputers has led to an increased need to display numbers in hexadecimal format (0–9, A–F). Standard drivers for this function are not available, so most applications are forced to use several packages to decode each digit of the display. Since 6 to 12 digits are often being displayed, this approach can become very expensive. This example demonstrates how the hexadecimal display format can be both decoded and the LED indicators driven using a single PAL for each digit of the display.

FUNCTIONAL DESCRIPTION

A hex decoder/lamp driver accepts a four-bit hex digit, converts it to its corresponding seven-segment display code, and activates the appropriate segments on the display. These drivers can be used in both direct-drive and multiplexed display applications. A single PAL can provide both the basic decode/drive functions, and additional useful features as well.

GENERAL DESCRIPTION

Figure 7.16.1 shows three digits of a display system that uses three PALs to implement the complete decoding and display-driving functions. The inputs to each section are a hex code on pins D_0–D_3, a ripple blanking signal, an intensity control signal, and a lamp test signal.

The hex codes are decoded to form the seven-segment patterns shown in Table 7.16.1. The input codes, digit represented, and segments driven are as follows:

TABLE 7.16.1. Function Description

D_3	D_2	D_1	D_0	Digit	Segments
0	0	0	0	0	ABCDEF
0	0	0	1	1	BC
0	0	1	0	2	ABDEG
0	0	1	1	3	ABCDG
0	1	0	0	4	BCFG
0	1	0	1	5	ACDFG
0	1	1	0	6	ACDEFG
0	1	1	1	7	ABC
1	0	0	0	8	ABCDEFG
1	0	0	1	9	ABCDFG
1	0	1	0	A	ABCEFG
1	0	1	1	B	CDEFG
1	1	0	0	C	ADEF
1	1	0	1	D	BCDEG
1	1	1	0	E	ADEFG
1	1	1	1	F	AEFG

Ripple-blanking input RBI is used to suppress leading zeroes in the display. The signal is propagated from the most significant digit to the least significant digit. If the digit input is zero and RBI is low (indicating that the previous digit is also zero), all segments are left blank and this digit position's ripple-blanking output RBO is set low.

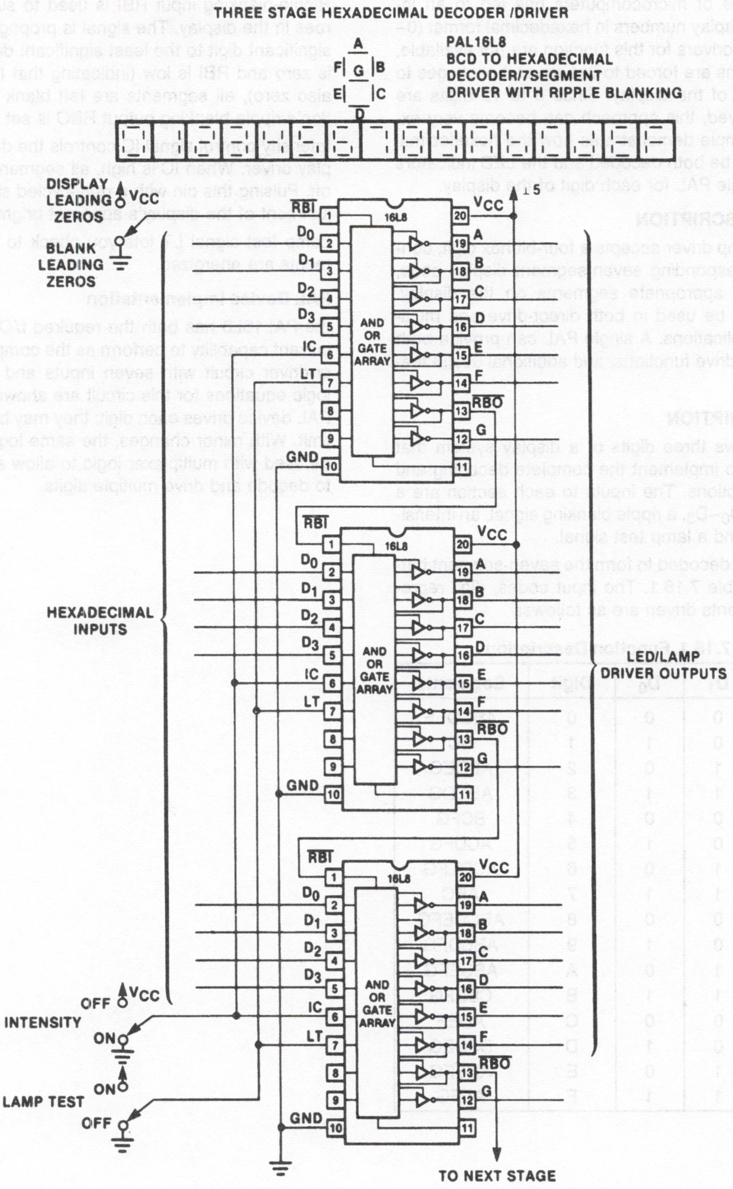
Intensity control signal IC controls the duty cycle of the display driver. When IC is high, all segment drivers are turned off. Pulsing this pin with a duty-cycled signal allows the adjustment of the display's apparent brightness.

Lamp test signal LT lets you check to see if all LED segments are energized.

PAL Device Implementation

the PAL16L8 has both the required I/O pins and the drive current capability to perform as the complete display decoder-driver circuit with seven inputs and eight outputs. The logic equations for this circuit are shown in the listing. One PAL device drives each digit; they may be cascaded without limit. With minor changes, the same logical structure could be used with multiplexer logic to allow a single PAL device to decode and drive multiple digits.

7.16 Hexadecimal 7-Segment Display Encoder (Continued)


THREE STAGE HEXADECIMAL DECODER /DRIVER
A
F
G
B
E
C
D
PAL16L8
BCD TO HEXADECIMAL
DECODER/7SEGMENT
DRIVER WITH RIPPLE BLANKING
DISPLAY LEADING ZEROS
VCC
BLANK LEADING ZEROS
RBI
D0
D1
D2
D3
IC
LT
GND
16L8
AND OR GATE ARRAY
VCC
RBO
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
1.5
HEXADECIMAL INPUTS
LED/LAMP DRIVER OUTPUTS
OFF
INTENSITY
ON
ON
LAMP TEST
OFF
TO NEXT STAGE

TL/L/9991-64

FIGURE 7.16.1. Hex Display Decoder-Driver Combinational Logic Diagram

7.16 Hexadecimal 7-Segment Display Encoder (Continued)

```
PALASM VERSION 1.5

PAL16L8
PAT07
HEX
BLANK
/RBI DO D1 D2 D3 IC LT NC NC GND
NC G /RBO F E D C B A VCC
IF(/IC)/A=/RBO*/DO*/D2+/RBO*/DO*D3+/RBO*D1*D2+
         /RBO*D1*D2*/D3+/RBO*DO*D2*/D3+/RBO*/D1*/D2*D3+LT
IF(IC)/B=/RBO*/D2*/D3+/RBO*/DO*/D2+/RBO*/DO*/D1*/D3+
         /RBO*DO*D1*/D3+/RBO*DO*/D1*/D3+LT
IF(IC)/C=/RBO*DO*/D1+/RBO*DO*/D2+/RBO*/D1*/D2+
         /RBO*D2*/D3+/RBO*/D2*D3+LT
IF(IC)/D=/RBO*/D1*D3+/RBO*/DO*/D2*/D3+
         /RBO*DO*D1*/D2+/RBO*/DO*D1*D2+/RBO*DO*/D1*D2+LT
IF(IC)/E=/RBO*/DO*/D2+/RBO*D2*D3+/RBO*/DO*D1+
         /RBO*D1*D3+LT
IF(IC)/F=/RBO*/DO*/D1+/RBO*/D2*D3+/RBO*D1*D3+
         /RBO*/DO*D2+/RBO*/D1*D2*/D3+LT
IF(VCC)RBO=/DO*/D1*/D2*/D3*/RBI
IF(/IC)/G=/RBO*D1*/D2+/RBO*DO*D3+/RBO*/D2*D3+
         /RBO*/DO*D1+/RBO*/D1*D2*/D3+LT
```

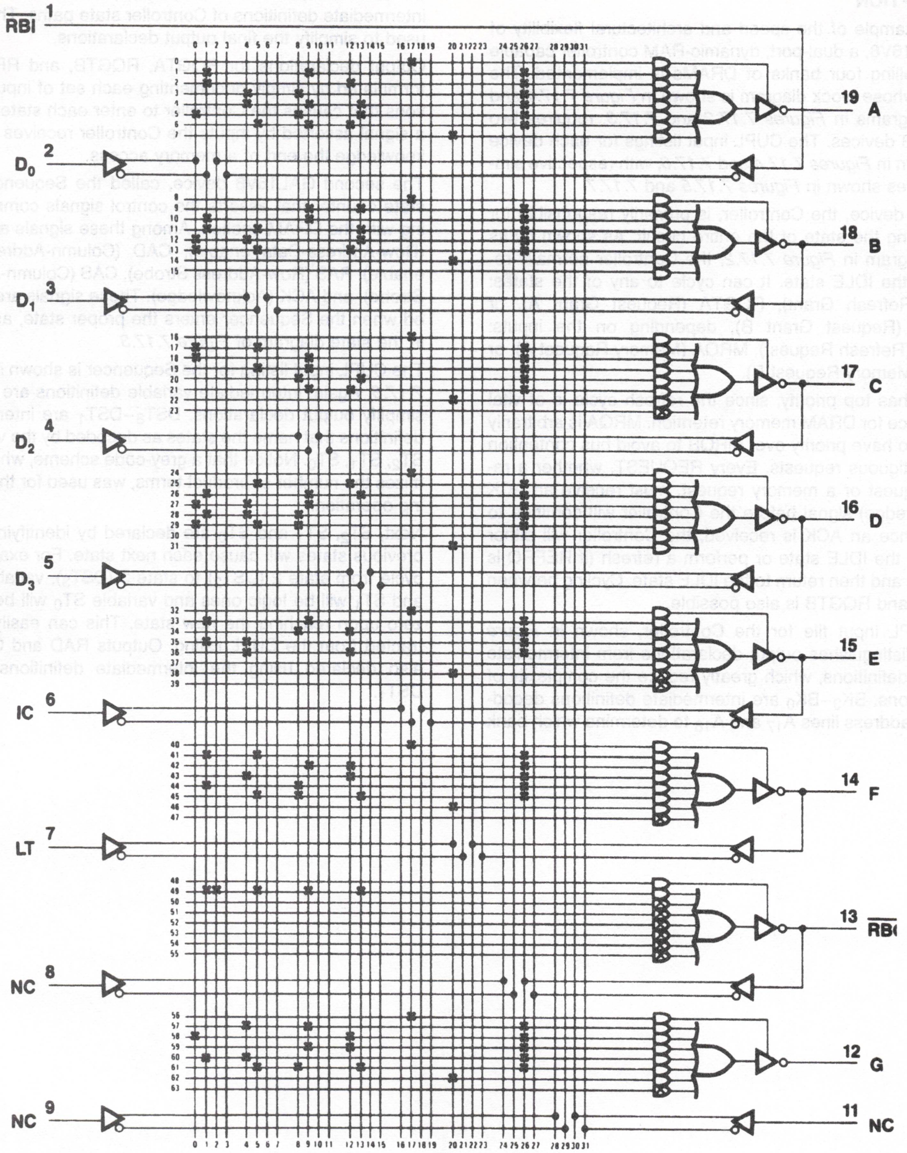
TL/L/9991-65

7.16 Hexadecimal 7-Segment Display Encoder (Continued)

```
HEX

                 11 1111 1111 2222 2222 2233
   0123 4567 8901 2345 6789 0123 4567 8901

BEG*FPLT PAL16L8    8
  0 ---- ---- ---- ---- -X-- ---- ---- ---- /IC
  1 -X-- ---- -X-- ---- ---- ---- --X- ---- /RBO*/DO*/D2
  2 -X-- ---- ---- X--- ---- ---- --X- ---- /RBO*/DO*D3
  3 ---- X--- X--- ---- ---- ---- --X- ---- /RBO*D1*D2
  4 ---- X--- X--- -X-- ---- ---- --X- ---- /RBO*D1*D2*/D3
  5 X--- ---- X--- -X-- ---- ---- --X- ---- /RBO*DO*D2*/D3
  6 ---- -X-- -X-- X--- ---- ---- --X- ---- /RBO*/D1*/D2*D3
  7 ---- ---- ---- ---- ---- X--- ---- ---- LT

  8 ---- ---- ---- ---- X--- ---- ---- ---- IC
  9 ---- ---- -X-- -X-- ---- ---- --X- ---- /RBO*/D2*/D3
 10 -X-- ---- -X-- ---- ---- ---- --X- ---- /RBO*/DO*/D2
 11 -X-- -X-- ---- -X-- ---- ---- --X- ---- /RBO*/DO*/D1*/D3
 12 X--- X--- ---- -X-- ---- ---- --X- ---- /RBO*DO*D1*/D3
 13 X--- -X-- ---- -X-- ---- ---- --X- ---- /RBO*DO*/D1*/D3
 14 ---- ---- ---- ---- ---- X--- ---- ---- LT
 15 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 16 ---- ---- ---- ---- X--- ---- ---- ---- IC
 17 X--- -X-- ---- ---- ---- ---- --X- ---- /RBO*DO*/D1
 18 X--- ---- -X-- ---- ---- ---- --X- ---- /RBO*DO*/D2
 19 ---- -X-- -X-- ---- ---- ---- --X- ---- /RBO*/D1*/D2
 20 ---- ---- X--- -X-- ---- ---- --X- ---- /RBO*D2*/D3
 21 ---- ---- -X-- X--- ---- ---- --X- ---- /RBO*/D2*D3
 22 ---- ---- ---- ---- ---- X--- ---- ---- LT
 23 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 24 ---- ---- ---- ---- X--- ---- ---- ---- IC
 25 ---- -X-- ---- X--- ---- ---- --X- ---- /RBO*/D1*D3
 26 -X-- ---- -X-- -X-- ---- ---- --X- ---- /RBO*/DO*/D2*/D3
 27 X--- X--- -X-- ---- ---- ---- --X- ---- /RBO*DO*D1*/D2
 28 -X-- X--- X--- ---- ---- ---- --X- ---- /RBO*/DO*D1*D2
 29 X--- -X-- X--- ---- ---- ---- --X- ---- /RBO*DO*/D1*D2
 30 ---- ---- ---- ---- ---- X--- ---- ---- LT
 31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 32 ---- ---- ---- ---- X--- ---- ---- ---- IC
 33 -X-- ---- -X-- ---- ---- ---- --X- ---- /RBO*/DO*/D2
 34 ---- ---- X--- X--- ---- ---- --X- ---- /RBO*D2*D3
 35 -X-- X--- ---- ---- ---- ---- --X- ---- /RBO*/DO*D1
 36 ---- X--- ---- X--- ---- ---- --X- ---- /RBO*D1*D3
 37 ---- ---- ---- ---- ---- X--- ---- ---- LT
 38 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 40 ---- ---- ---- ---- X--- ---- ---- ---- IC
 41 -X-- -X-- ---- ---- ---- ---- --X- ---- /RBO*/DO*/D1
 42 ---- ---- -X-- X--- ---- ---- --X- ---- /RBO*/D2*D3
 43 ---- X--- ---- X--- ---- ---- --X- ---- /RBO*D1*D3
 44 -X-- ---- X--- ---- ---- ---- --X- ---- /RBO*/DO*D2
 45 ---- -X-- X--- -X-- ---- ---- --X- ---- /RBO*/D1*D2*/D3
 46 ---- ---- ---- ---- ---- X--- ---- ---- LT
 47 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 48 ---- ---- ---- ---- ---- ---- ---- ----
 49 -XX- -X-- -X-- -X-- ---- ---- ---- ---- /DO*/D1*/D2*/D3*/RBI
 50 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 51 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 52 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 53 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 54 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 55 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 56 ---- ---- ---- ---- -X-- ---- ---- ---- /IC
 57 ---- X--- -X-- ---- ---- ---- --X- ---- /RBO*D1*/D2
 58 X--- ---- ---- X--- ---- ---- --X- ---- /RBO*DO*D3
 59 ---- ---- -X-- X--- ---- ---- --X- ---- /RBO*/D2*D3
 60 -X-- X--- ---- ---- ---- ---- --X- ---- /RBO*/DO*D1
 61 ---- -X-- X--- -X-- ---- ---- --X- ---- /RBO*/D1*D2*/D3
 62 ---- ---- ---- ---- ---- X--- ---- ---- LT
 63 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

END*FPLT

 LEGEND:  X : FUSE NOT BLOWN (L,N,O)   - : FUSE BLOWN   (H,P,1)

 NUMBER OF FUSES BLOWN = 1496
```

TL/L/9991-69

7.16 Hexadecimal 7-Segment Display Encoder (Continued)


RBI 1
D0 2
D1 3
D2 4
D3 5
IC 6
LT 7
NC 8
NC 9
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0–63
19 A
18 B
17 C
16 D
15 E
14 F
13 RBI
12 G
11 NC

TL/L/9991–72

FIGURE 7.16.2. PAL16L8 Logic Diagram Showing Lamp Driver Pattern

7.17 Dual-Port RAM Controller

DESCRIPTION

As an example of the speed and architectural flexibility of the GAL16V8, a dual-port, dynamic-RAM controller capable of controlling four banks of DRAMs is implemented. The design, whose block diagram is shown in *Figure 7.17.1* and state diagrams in *Figures 7.17.2* and *7.17.3*, requires two GAL16V8 devices. The CUPL input listings for each device are shown in *Figures 7.17.4* and *7.17.6*, with respective simulation files shown in *Figures 7.17.5* and *7.17.7*.

The first device, the Controller, is primarily responsible for maintaining the state of the entire circuit. As shown by its state diagram in *Figure 7.17.2*, the Controller normally resides in the IDLE state. It can cycle to any of the states: RFGT (Refresh Grant), RQGTA (Request Grant A), or RQGTB (Request Grant B), depending on the inputs: REFRQ (Refresh Request), MRQA (Memory Request A), or MRQB (Memory Request B).

REFRQ has top priority, since the refresh cycle is of vital importance for DRAM memory retention. MRQA is arbitrarily chosen to have priority over MRQB to avoid bus contention with contiguous requests. Every REQUEST, whether a refresh request or a memory request, must receive an ACK (acknowledge) signal before the Controller will continue to cycle. Once an ACK is received, the Controller will either return to the IDLE state or perform a refresh (if REFRQ is present), and then return to the IDLE state. Cycling between RQGTA and RQGTB is also possible.

The CUPL input file for the Controller, shown in *Figure 7.17.4*, distinguishes output declarations from intermediate variable definitions, which greatly reduce the complexity of declarations. BK_3–BK_0 are intermediate definitions decoded from address lines A_{17} and A_{16} to determine which bank will be selected. RQGTAS, RQGTBS, and RFGTS are also intermediate definitions of Controller state paths. These are used to simplify the final output declarations.

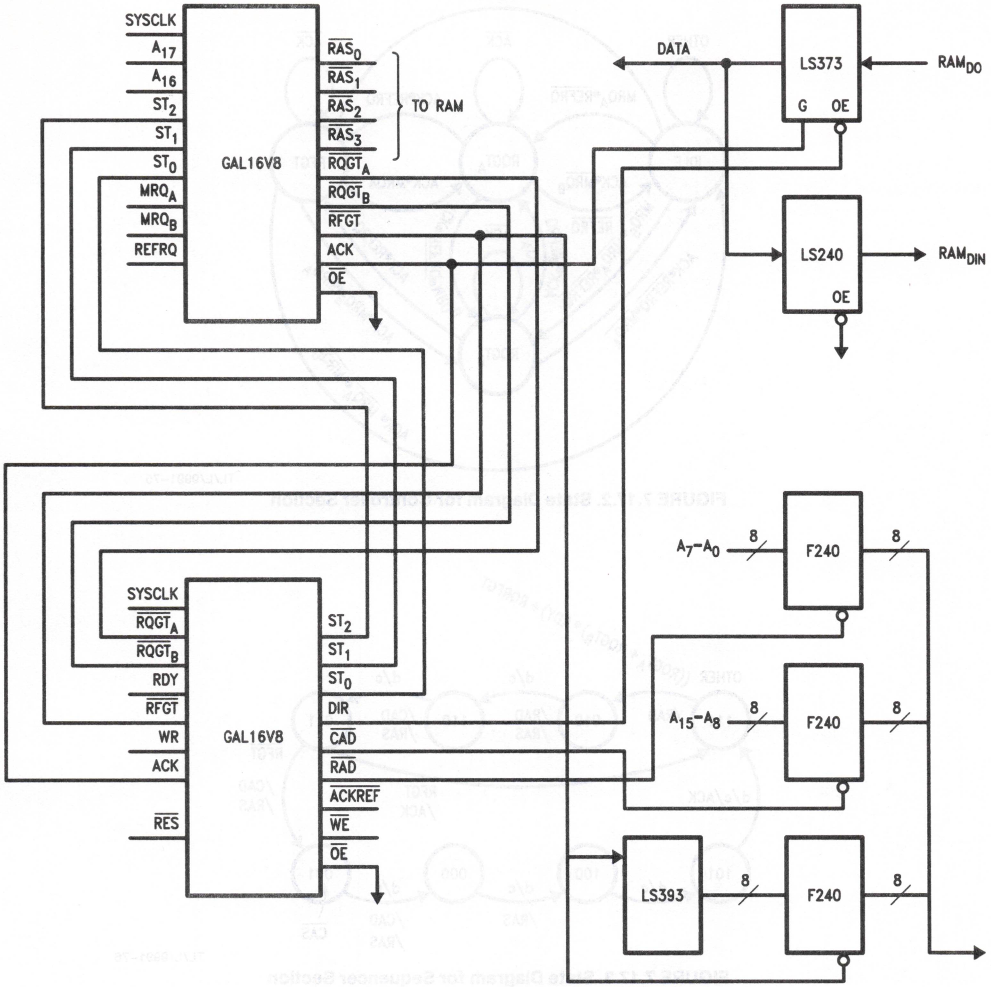
Output declarations for RQGTA, RQGTB, and RFTG are formulated by simply documenting each set of input conditions that causes the Controller to enter each state. ACK is a signal asserted by inputs the Controller receives that acknowledge the end of a memory access.

The second GAL16V8 device, called the Sequencer, is a state counter that asserts the control signals communicating with the DRAM section. Among these signals are: RAD (Row-Address-Data enable), CAD (Column-Address-Data enable), RAS (Row-Address Strobe), CAS (Column-Address Strobe), and ACK (Acknowledge). These signals are asserted when the Sequencer enters the proper state, as shown in the state diagram of *Figure 7.17.3*.

The CUPL input listing for the Sequencer is shown in *Figure 7.17.6*. Again, intermediate variable definitions are used to simplify output declarations. DST_8–DST_1 are intermediate definitions that name the states as decoded by the variables ST_2, ST_1, ST_0. Notice that a grey-code scheme, which minimizes the number of product terms, was used for the counting operation.

Next, ST_2, ST_1 and ST_0 are declared by identifying which previous states will cause each next state. For example, to cycle from state 2 (DST_2) to state 3 (DST_3), variables ST_2 and ST_1 will be logic ones and variable ST_0 will be a logic zero upon reaching the new state. This can easily be extracted from the CUPL listing. Outputs RAD and CAD are also declared using the intermediate definitions DST_8–DST_1.

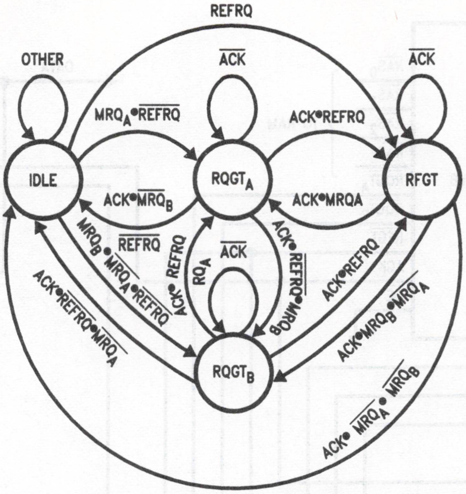
7.17 Dual-Port RAM Controller (Continued)


SYSCLK
A_{17}
A_{16}
ST_2
ST_1
ST_0
MRQ_A
MRQ_B
REFRQ
GAL16V8
$\overline{RAS_0}$
$\overline{RAS_1}$
$\overline{RAS_2}$
$\overline{RAS_3}$
TO RAM
$\overline{RQGT_A}$
$\overline{RQGT_B}$
$\overline{RFGT}$
ACK
$\overline{OE}$
DATA
LS373
RAM_{DO}
G
OE
LS240
RAM_{DIN}
OE
SYSCLK
$\overline{RQGT_A}$
$\overline{RQGT_B}$
RDY
$\overline{RFGT}$
WR
ACK
$\overline{RES}$
GAL16V8
ST_2
ST_1
ST_0
DIR
$\overline{CAD}$
$\overline{RAD}$
$\overline{ACKREF}$
$\overline{WE}$
$\overline{OE}$
A_7–A_0
8
F240
8
A_{15}–A_8
8
F240
8
LS393
8
F240
8

TL/L/9991-74

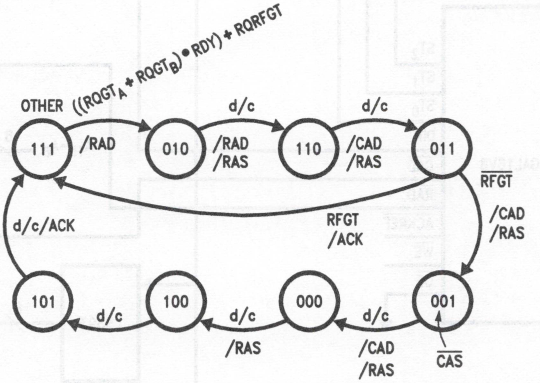
FIGURE 7.17.1. Block Diagram

7.17 Dual-Port RAM Controller (Continued)


REFRQ
OTHER
$\overline{ACK}$
$\overline{ACK}$
IDLE
$RQGT_A$
RFGT
$RQGT_B$
$MRQ_A \bullet \overline{REFRQ}$
$ACK \bullet REFRQ$
$ACK \bullet \overline{MRQ_B}$
$ACK \bullet MRQA$
$\overline{REFRQ}$
$MRQ_B \bullet \overline{MRQ_A} \bullet \overline{REFRQ}$
$ACK \bullet REFRQ$
RQ_A
$\overline{ACK}$
$ACK \bullet \overline{REFRQ} \bullet MRQ_B$
$ACK \bullet REFRQ$
$ACK \bullet \overline{REFRQ} \bullet MRQ_A$
$ACK \bullet MRQ_B \bullet \overline{MRQ_A}$
$ACK \bullet \overline{MRQ_A} \bullet \overline{MRQ_B}$

TL/L/9991-75

FIGURE 7.17.2. State Diagram for Controller Section


OTHER
$((RQGT_A + RQGT_B) \bullet RDY) + RQRFGT$
111
/RAD
d/c
010
/RAD
/RAS
d/c
110
/CAD
/RAS
011
$\overline{RFGT}$
/CAD
/RAS
RFGT
/ACK
d/c/ACK
101
d/c
100
d/c
/RAS
000
d/c
/CAD
/RAS
001
$\overline{CAS}$

TL/L/9991-76

FIGURE 7.17.3. State Diagram for Sequencer Section

7.17 Dual-Port RAM Controller (Continued)

The two GAL16V8-25 devices can be clocked at cycle times as fast as 35 ns (28.5 MHz), ample enough for the tight timings required to run a DRAM at its specified access times. The GAL16V8's power-up reset feature comes in handy in this circuit, since no inputs were available for a reset term. To test the functionality of this circuit, the simulation facilities of CUPL were used.

It should be noted that although the Controller uses all eight registers in the device, the Sequencer requires seven registers and one combinational output. While the Controller could be implemented in a traditional PAL configuration (16R8), the Sequencer requires a nonstandard architecture which can only be implemented in a GAL16V8 device. This is one of the biggest advantages of GAL devices—the flexibility of the architecture.

```
/*******************************************************************/
/*                                                                 */
/*                        CUPL INPUT FILE                          */
/*        Design input for the controller section of the           */
/*                   Dual Port DRAM Controller                     */
/*                                                                 */
/*******************************************************************/
/*             Allowable Target Device Types: GAL16V8              */
/*******************************************************************/

                    PARTNO    CONTROLLER SECTION;
                    NAME      DRAM CONTROLLER;
                    DATE      03/28/86 ;
                    REV       01 ;
                    DESIGNER  Joe Engineer;
                    COMPANY   National Semiconductor;
                    ASSEMBLY  ONE;
                    LOCATION  U10;

/** Inputs   **/

PIN 1          =  SYSCLK;
PIN [2,3]      = [A16,A17] ;
PIN [4..6]     = [ST2,ST1,ST0] ;
PIN 7          = MRQA ;
PIN 8          = MRQB ;
PIN 9          = REFRQ ;
PIN 11         = !OE ;

/** Outputs  **/

PIN 19         = !RAS0 ;
PIN 18         = !RAS1 ;
PIN 17         = !RAS2 ;
PIN 16         = !RAS3 ;
PIN 15         = !RQGTA ;
PIN 14         = !RQGTB ;
PIN 13         = !RFGT ;
PIN 12         = ACK ;
```

TL/L/9991-77

FIGURE 7.17.4. Design Input File for Controller Section

7.17 Dual-Port RAM Controller (Continued)

```
/** Declarations and Intermediate Variable Definitions **/

BK0 = (!A17 & !A16) # RFGT ;
BK1 = (!A17 & A16) # RFGT ;
BK2 = (A17 & !A16) # RFGT ;
BK3 = (A17 & A16) # RFGT ;

RASEN = !ST2 & ST1 & !ST0 # ST2 & ST1 & !ST0 # !ST2 & ST1 & ST0 #
        !ST2 & !ST1 & ST0 # !ST2 & !ST1 & !ST0 ;

RAS0.D = BK0 & RASEN ;
RAS1.D = BK1 & RASEN ;
RAS2.D = BK2 & RASEN ;
RAS3.D = BK3 & RASEN ;

RQGTAS = RQGTA & !RQGTB & !RFGT ;

RQGTBS = !RQGTA & RQGTB & !RFGT ;

RFGTS = !RQGTA & !RQGTB & RFGT ;

IDLE = !RQGTA & !RQGTB & !RFGT ;

RQGTA.D = (IDLE & MRQA & !REFRQ # RQGTAS & !ACK # RQGTBS & ACK &
          MRQA & !REFRQ   # RFGTS & ACK & MRQA) & !(ACK & !MRQA &
          !MRQB & !REFRQ) ;

RQGTB.D = (IDLE & !MRQA & !REFRQ & MRQB # RQGTBS & !ACK # RQGTAS &
          ACK & MRQB & !REFRQ   # RFGTS & ACK & !MRQA & MRQB) & !(ACK &
          !MRQA & !MRQB & !REFRQ) ;

RFGT.D = (IDLE & REFRQ # RFGTS & !ACK # RQGTAS & ACK & REFRQ #
          RQGTBS & ACK & REFRQ) & !(ACK & !MRQA & !MRQB & !REFRQ) ;

ACK.D = ST2 & !ST1 & ST0 # !ST2 & ST1 & ST0 & RFGT ;
```

TL/L/9991-78

FIGURE 7.17.4. Design Input File for Controller Section (Continued)

```
/****************************************************************/
/*                                                              */
/*                     CUPL SIMULATION FILE                     */
/*      Simulation input for the controller section of the     */
/*                  Dual Port DRAM Controller                   */
/*                                                              */
/****************************************************************/
/*            Allowable Target Device Types: GAL16V8            */
/****************************************************************/

                PARTNO     CONTROLLER SECTION;
                NAME       DRAM CONTROLLER;
                DATE       03/28/86 ;
                REV        01 ;
                DESIGNER   Joe Engineer;
                COMPANY    National Semiconductor;
                ASSEMBLY   ONE;
                LOCATION   U10;

ORDER:
SYSCLK,%2,A17,A16,%2,ST2,ST1,ST0,%2,MRQA,MRQB,REFRQ,%2,!OE,%4,
!RAS0,!RAS1,!RAS2,!RAS3,%2,!RQGTA,!RQGTB,!RFGT,%2,ACK;

VECTORS:
$msg" S                            !!     ";
$msg" Y             R        !!!!  RR!    ";
$msg" S           MME        RRRR  QQR    ";
$msg" C  AA  SSS  RRF   !    AAAA  GGF  A ";
$msg" L  11  TTT  QQR   O    SSSS  TTG  C ";
$msg" K  76  210  ABQ   E    0123  ABT  K ";
$msg" ----------------------------------- ";

      0  00  101  000   0    XXXX  XXX  X
      C  00  101  000   0    HHHH  XXX  H
      C  00  111  000   0    HHHH  HHH  L
      C  00  111  000   0    HHHH  HHH  L
      C  00  111  100   0    HHHH  LHH  L
      C  00  010  100   0    LHHH  LHH  L
      C  00  110  100   0    LHHH  LHH  L
      C  00  011  100   0    LHHH  LHH  L
      C  00  001  100   0    LHHH  LHH  L
      C  00  000  100   0    LHHH  LHH  L
      C  00  100  100   0    HHHH  LHH  L
      C  00  101  110   0    HHHH  LHH  H
      C  00  111  110   0    HHHH  HLH  L
      C  11  111  010   0    HHHH  HLH  L
      C  11  111  010   0    HHHH  HLH  L
      C  11  010  010   0    HHHL  HLH  L
      C  11  110  010   0    HHHL  HLH  L
      C  11  011  010   0    HHHL  HLH  L
      C  11  001  010   0    HHHL  HLH  L
      C  11  000  000   0    HHHL  HLH  L
      C  11  100  101   0    HHHH  HLH  L
      C  11  101  101   0    HHHH  HLH  H
      C  00  111  101   0    HHHH  HHL  L
      C  00  111  101   0    HHHH  HHL  L
      C  11  010  000   0    LLLL  HHL  L
      C  11  110  000   0    LLLL  HHL  L
      C  11  011  000   0    LLLL  HHL  H
      C  11  001  000   0    LLLL  HHH  L
      C  11  000  000   0    HHHL  HHH  L
      C  11  100  101   0    HHHH  HHL  L
      C  11  101  101   0    HHHH  HHL  H
      C  00  111  101   0    HHHH  LHH  L
      C  00  111  101   0    HHHH  LHH  L
```

TL/L/9991-80

FIGURE 7.17.5. Simulation File for Controller Section

7.17 Dual-Port RAM Controller (Continued)

```
/******************************************************************/
/*                                                                */
/*                       CUPL INPUT FILE                          */
/*      Design input for the sequencer section for the            */
/*                  Dual Port DRAM Controller                     */
/*                                                                */
/******************************************************************/
/*     Allowable Target Device Types: GAL16V8                     */
/******************************************************************/

                  PARTNO     SEQUENCER SECTION;
                  NAME       DRAM CONTROLLER;
                  DATE       03/28/86 ;
                  REV        01 ;
                  DESIGNER   Joe Engineer;
                  COMPANY    National Semiconductor;
                  ASSEMBLY   TWO;
                  LOCATION   U11;

/**  Inputs  **/

PIN 1          =  SYSCLK;
PIN [2,3]      = [!RQGTA,!RQGTB] ;
PIN 4          = RDY ;
PIN 5          = !RFGT ;
PIN 6          = !WR ;
PIN 7          = ACK ;
PIN 8          = !RES ;
PIN 11         = !OE ;
/**  Outputs  **/

PIN 19         = ST2 ;
PIN 18         = ST1 ;
PIN 17         = ST0 ;
PIN 16         = DIR ;
PIN 15         = !CAD ;
PIN 14         = !RAD ;
PIN 13         = !ACKREF ;
PIN 12         = !WE ;
```

TL/L/9991-81

FIGURE 7.17.6. Input File for Sequencer Section

7.17 Dual-Port RAM Controller (Continued)

```
/** Declarations and Intermediate Variable Definitions **/

DST1 = ST2 & ST1 & ST0 ;
DST2 = !ST2 & ST1 & !ST0 ;
DST3 = ST2 & ST1 & !ST0 ;
DST4 = !ST2 & ST1 & ST0 ;
DST5 = !ST2 & !ST1 & ST0 ;
DST6 = !ST2 & !ST1 & !ST0 ;
DST7 = ST2 & !ST1 & !ST0 ;
DST8 = ST2 & !ST1 & ST0 ;

STCYC = ((RQGTA # RQGTB) & RDY # RFGT) ;

ST2.D = (DST2 # DST6 # DST8 # DST7 # DST4 & RFGT) # RES #
        DST1 & !STCYC ;

ST1.D = DST2 # DST3 # DST8 # DST4 & RFGT # DST1 # RES ;

ST0.D = (DST3 # DST4 # DST7 # DST8) # RES # DST1 & !STCYC ;

DIR.D = WR & !DST1 ;

CAD.D = DST3 & !RFGT # DST4 & !RFGT # DST5 ;

RAD.D = (RQGTA # RQGTB) & RDY & (DST1 # DST2) ;

ACKREF = RFGT & ACK ;

WE.D = WR & (DST5 # DST6) ;
```

TL/L/9991-82

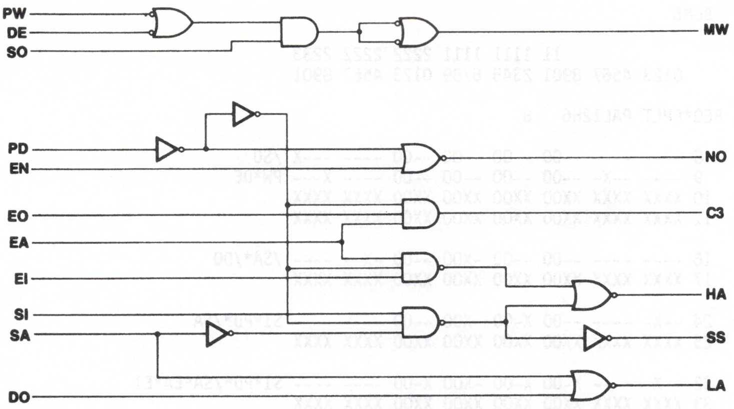
FIGURE 7.17.6. Input File for Sequencer Section (Continued)

```
/**************************************************************/
/*                                                            */
/*                  CUPL SIMULATION FILE                      */
/*      Simulation File for the sequencer section of the      */
/*                Dual Port DRAM Controller                   */
/*                                                            */
/**************************************************************/
/*          Allowable Target Device Types: GAL16V8            */
/**************************************************************/

                PARTNO    SEQUENCER SECTION;
                NAME      DRAM CONTROLLER;
                DATE      03/28/86 ;
                REV       01 ;
                DESIGNER  Joe Engineer;
                COMPANY   National Semiconductor;
                ASSEMBLY  TWO;
                LOCATION  U11;
ORDER:
SYSCLK,%2,!RES,%2,!RQGTA,!RQGTB,!RFGT,%2,RDY,!WR,ACK,%2,!OE,%4,
ST2,ST1,ST0,%2,DIR,%2,!CAD,!RAD,%2,!WE ;

VECTORS:

$num" S     !!                           ";
$num" Y     RR!                          ";
$num" S  !  QQR                    !!    ";
$num" C  R  GGF  R!A  !    SSS  D  CR  ! ";
$num" L  E  TTG  DWC  O    TTT  I  AA  W ";
$num" K  S  ABT  YRK  E    210  R  DD  E ";
$num" -----------------------------------";
      0  0  111  010  0    XXX  X  XX  X
      0  0  111  010  0    XXX  X  XX  X
      C  0  111  010  0    HHH  L  XH  H
      C  1  011  010  0    HHH  L  HH  H
      C  1  011  110  0    LHL  L  HL  H
      C  1  011  110  0    HHL  L  HL  H
      C  1  011  010  0    LHH  L  LH  H
      C  1  011  010  0    LLH  L  LH  H
      C  1  011  010  0    LLL  L  LH  H
      C  1  011  010  0    HLL  L  HH  H
      C  1  011  010  0    HLH  L  HH  H
      C  1  011  010  0    HHH  L  HH  H
      C  1  011  010  0    HHH  L  HH  H
      C  1  011  010  0    HHH  L  HH  H
```

TL/L/9991-83

FIGURE 7.17.7. Simulation File for Sequencer Section

7.18 8086 CPU Board Random Control Logic


PW
DE
SO
MW
PD
EN
NO
EO
C3
EA
EI
HA
SI
SS
SA
LA
DO

TL/L/9991-84

FIGURE 7.18.1 Control Logic for 8086 CPU Board

```
PALASM VERSION 1.5

PAL12H6
PAT03
8086
CPU
PD EN EO EA S1 SA E1 DO DE GND
SO NC3 NO C3 HA SS LA MW PW VCC
MW=/SO+PW*DE
LA=/SA*/DO
SS=S1*PD*/SA
HA=S1*PD*/SA*EA*E1
C3=PD*EO*EA
NO=PD*/EN
```

TL/L/9991-85

7.18 8086 CPU Board Random Control Logic (Continued)

```
 8086


                 11 1111 1111 2222 2222 2233
    0123 4567 8901 2345 6789 0123 4567 8901

BEG*FPLT PAL12H6   8

  8 ---- ---- --00 --00 --00 --00 ---- ---X /S0
  9 ---- --X- --00 --00 --00 --00 ---- X--- PW*DE
 10 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX
 11 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

 16 ---- ---- --00 --00 -X00 --00 -X-- ---- /SA*/DO
 17 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

 24 --X- ---- --00 X-00 -X00 --00 ---- ---- S1*PD*/SA
 25 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

 32 --X- ---- X-00 X-00 -X00 X-00 ---- ---- S1*PD*/SA*EA*E1
 33 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

 40 --X- X--- X-00 --00 --00 --00 ---- ---- PD*EO*EA
 41 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX

 48 -XX- ---- --00 --00 --00 --00 ---- ----  PD*/EN
 49 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX
 50 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX
 51 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX


END*FPLT

 LEGEND:  X : FUSE NOT BLOWN (L,N,0)    - : FUSE BLOWN   (H,P,1)
          0 : PHANTOM FUSE   (L,N,0)    0 : PHANTOM FUSE (H,P,1)

 NUMBER OF FUSES BLOWN =  206
```

TL/L/9991-89

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU*

PAL DEVICES FOR EASY INTERFACE BETWEEN DP8408/09** DRAM CONTROLLER AND POPULAR MICROPROCESSORS

High storage density and low cost have made dynamic RAMs the designers choice in most memory applications. However, the major drawbacks of dynamic RAMs are the complex timing involved and periodic refresh needed to keep all memory cells charged. With the introduction of the DP8408/09 Dynamic RAM controller/driver, the above complexities are simplified.

Use of PAL devices adds flexibility in the design as PAL device logic equations can be modified by the user for his/her application and programmed into any of the PAL devices. In addition, PAL devices lower the parts count in memory system design. For most memory operations, the PAL devices (DP84312/322/332) can be directly connected between the control signals from the CPU chip set and the DP8408/09 dynamic RAM controller. The PAL device allows hidden refresh using the DP8408/09. In a standard memory cycle, the access can be slowed by one clock cycle to accommodate slower memories. This extra wait state will not appear during the hidden refresh cycle, so faster devices on the CPU bus will not be affected. Similarly, PAL devices allow for the insertion of wait states for processors operating at high CPU clock frequencies to use slower dynamic RAMs.

The following three applications describe the use of National's PAL16R6, PAL16R4 and PAL16R8 for the ease and flexibility of interfacing DP8408/09 with popular microprocessors such as the NS32032, 68000, 8086, and 8088. Today the PAL device family offers the designer flexibility to design desired speed/power PAL devices in his memory systems, and achieve the memory operations at very high frequencies with or without wait state conditions.

GENERAL DESCRIPTION

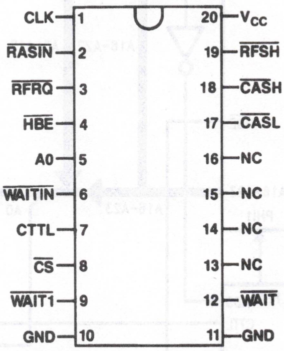
The DP84312 dynamic RAM Controller interface is a PAL device for interface between the DP8409 dynamic RAM Controller and the NS32032 microprocessor.

Using timing signals from the NS32032 timing and control unit and the NS32032 the DP84312 supplies all control signals needed to perform memory read, write, byte write, and refresh.

FEATURES

- Low parts count memory system
- Allows the DP8409 to perform hidden refresh
- Allows for the insertion of wait states for slow dynamic RAMs
- Supplies independent $\overline{\text{CAS}}$s for byte writing
- Possibility of operation at 8 MHz with no wait states
- 20-pin 0.3 inch wide package
- Standard National Semiconductor PAL device part (PAL16R6)
- PAL device logic equations can be modified by the user for his/her specific application and programmed into any of the National Semiconductor PAL device family, including the new high speed PAL devices

Dual-In-Line Package


CLK 1
$\overline{\text{RASIN}}$ 2
$\overline{\text{RFRQ}}$ 3
$\overline{\text{HBE}}$ 4
A0 5
$\overline{\text{WAITIN}}$ 6
CTTL 7
$\overline{\text{CS}}$ 8
$\overline{\text{WAIT1}}$ 9
GND 10
20 V_{CC}
19 $\overline{\text{RFSH}}$
18 $\overline{\text{CASH}}$
17 $\overline{\text{CASL}}$
16 NC
15 NC
14 NC
13 NC
12 $\overline{\text{WAIT}}$
11 GND

TL/L/9991-90

Top View

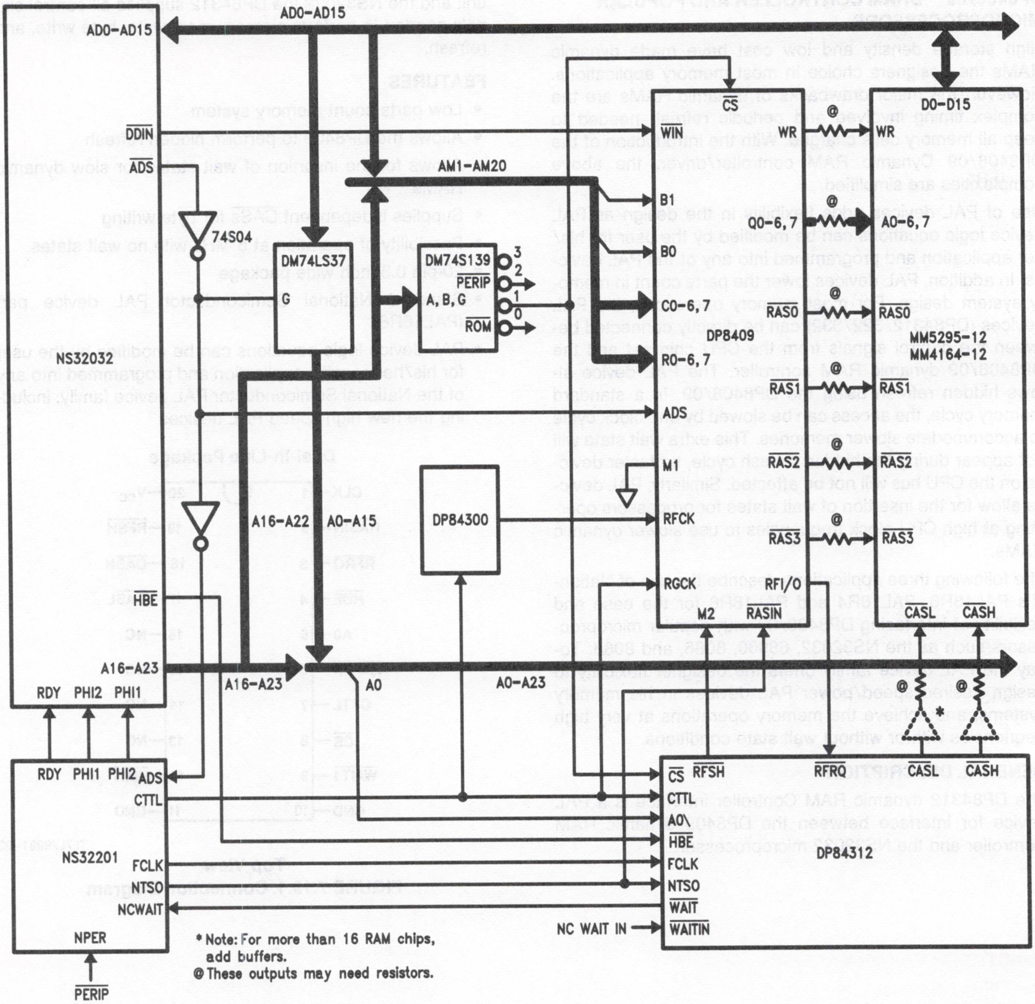
FIGURE 7.19.1. Connection Diagram

---

*Applications contained in this section are for illustration purposes only and National makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

**DP8408/09 is part of the interface product line at National Semiconductor Corporation.

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)


AD0–AD15
AD0–AD15
CS
D0–D15
DDIN
WIN
WR
@
WR
ADS
AM1–AM20
B1
74S04
Q0–6, 7
@
A0–6, 7
DM74LS37
DM74S139
3
2
1
0
B0
PERIP
G
A, B, G
ROM
C0–6, 7
RAS0
@
RAS0
DP8409
MM5295–12
MM4164–12
NS32032
R0–6, 7
RAS1
@
RAS1
ADS
M1
RAS2
@
RAS2
DP84300
A16–A22
A0–A15
RFCK
RAS3
@
RAS3
HBE
RGCK
RFI/O
M2
RASIN
CASL
CASH
A16–A23
A16–A23
A0
A0–A23
RDY
PHI2
PHI1
@
@
*
RDY
PHI1
PHI2
ADS
CS
RFSH
RFRQ
CASL
CASH
CTTL
CTTL
A0
HBE
NS32201
FCLK
FCLK
DP84312
NTSO
NTSO
NCWAIT
WAIT
NC WAIT IN
WAITIN
NPER
PERIP

* Note: For more than 16 RAM chips, add buffers.

@ These outputs may need resistors.

TL/L/9991-91

FIGURE 7.19.2. System Block Diagram

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)

MNEMONIC DESCRIPTION

Input Signals

CLK	Clock input. This clock comes from the FCLK output of the NS32201 timing and control unit, and supplies timing for the internal logic.
$\overline{\text{RASIN}}$	$\overline{\text{RAS}}$ input. This input is connected to the NTSO pin of the NS32201. This signal marks the start of a memory cycle.
$\overline{\text{RFRQ}}$	Refresh request. The DP8409 requests a forced refresh with this input.
$\overline{\text{HBE}}$, A0	Address select inputs. These inputs select the type of write during a write cycle, and select their respective $\overline{\text{CAS}}$ outputs. These inputs must remain stable throughout the memory cycle.
$\overline{\text{WAITIN}}$	This wait input allows other devices to use the NCWAIT line of the NS16201 clock chip.
CTTL	System clock input. This clock is used to synchronize the memory system to the microprocessor clock.
$\overline{\text{CS}}$	Chip select. This input is used to determine if a memory cycle or a hidden refresh cycle is to be performed.
$\overline{\text{WAIT1}}$	Insert one wait state. This input allows the use of slow memories with a microprocessor using a fast clock by inserting a wait state in selected memory cycles.

Output Signals

$\overline{\text{RFSH}}$	Refresh. This output switches the DP8409 to a refresh mode.
$\overline{\text{CASH}}$, $\overline{\text{CASL}}$	$\overline{\text{CAS}}$ outputs. $\overline{\text{CASH}}$ is for controlling the high bank of dynamic RAMs, while $\overline{\text{CASL}}$ controls the $\overline{\text{CAS}}$ line of the lower bank of RAMs. If only eight RAMs are used in each bank, the $\overline{\text{CAS}}$ outputs will directly drive the memories. For large arrays, these outputs should be buffered with a high current driver, such as the DP84244 MOS driver.
$\overline{\text{WAIT}}$	This output controls the insertion of wait states. This output is ORed with $\overline{\text{WAITIN}}$ to allow other devices to insert wait states.

FUNCTIONAL DESCRIPTION

The DP84312 detects the start of a memory cycle when NTSO from the NS32032 timing and control unit (TCU) goes low. The NTSO signal is also used to supply $\overline{\text{RASIN}}$ to the DP8409 dynamic RAM controller. After the DP8409 has latched the row address and supplied the column address to the DRAMs, the DP84312 latches the column address. The DP84312 supplies two $\overline{\text{CAS}}$ outputs: one for the high byte of memory, and the other for the low byte. The ability to control the upper and lower bytes of memory separately is important during a memory write cycle where one byte of memory is to be written (byte write).

By connecting $\overline{\text{WAIT1}}$ of the DP84312 to ground, all selected memory cycles will have one wait state inserted. This allows an NS32032 operating at high CPU clock frequency to use slower dynamic RAMs.

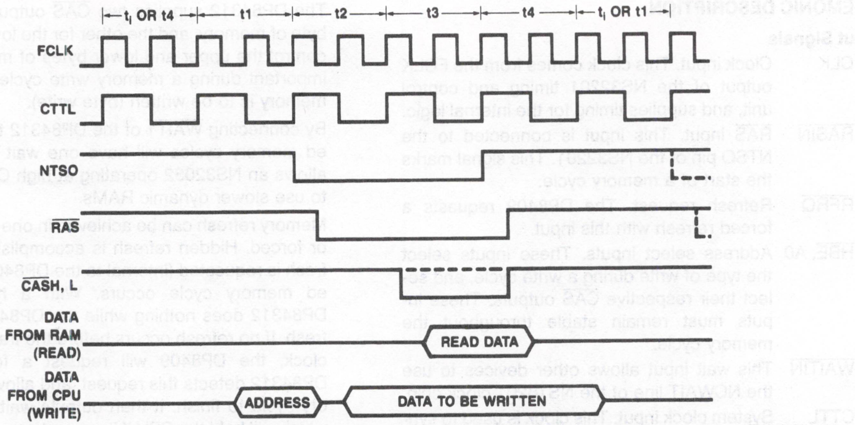
Memory refresh can be achieved in one of two ways: hidden or forced. Hidden refresh is accomplished whenever a refresh is requested (internal to the DP8409) and an unselected memory cycle occurs. With a hidden refresh, the DP84312 does nothing while the DP8409 performs the refresh. If no refresh occurs before the trailing edge of refresh clock, the DP8409 will request a forced refresh. The DP84312 detects this request, and allows the current memory cycle to finish. It then outputs wait states to the CPU, which will hold the CPU if it requests a memory cycle. During this time the DP84312 has switched the dynamic RAM controller to the auto refresh mode, allowing it to perform a refresh. At the end of the refresh cycle, the DP8409 is switched back to the auto access mode, and the wait is removed after a sufficient $\overline{\text{RAS}}$ precharge time. The total forced refresh takes four CPU clock cycles, of which some, none or all may be actual wait states. If the CPU does not request a memory cycle during this refresh cycle, the refresh will not impact the CPU's performance.

The DP84312 can possibly be operated at 8 MHz with no wait states (WAIT1 = "1") given the following conditions:

T2 + T3 = 250 ns
NTSO generation = 15 ns max
$\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ delay DP8409-2 = 130 ns max
External $\overline{\text{CASH, L}}$ generation using 74S02 and 74S240
    7.5 ns (74S02) + 10 ns (74S240) − 7.5 ns (less load on 8409 $\overline{\text{CAS}}$ line) = 10 ns max
Transceiver delay = 12 ns max
NS16032 data setup = 20 ns max
$\therefore$ Minimum t_{CAC} = 63 ns
    = 250 − 15 − 130 − 10 − 12 − 20
Minimum t_{RAS} = 250 ns
Minimum t_{RP} = 250 ns
Minimum t_{RAH} = 20 ns

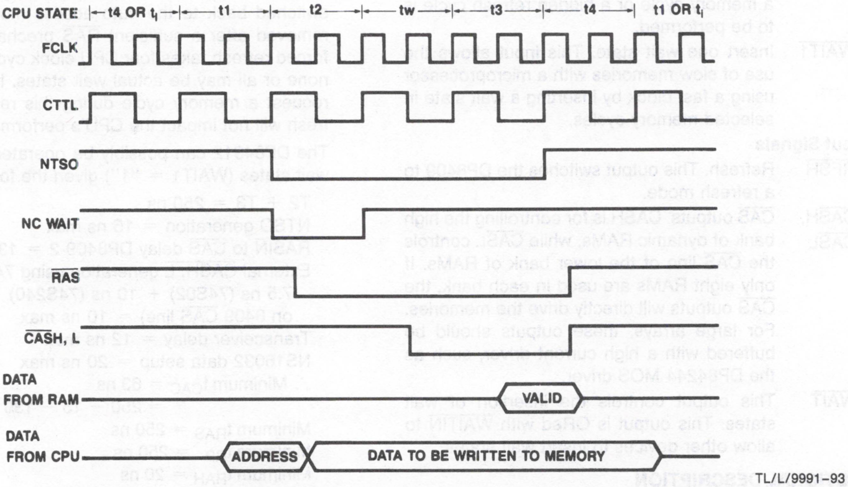
The DP84312 is a standard National PAL device part (PAL 16R6). The user can modify the PAL device equations to support his/her particular application. The DP84312 logic equations, function table (functional test), and logic diagram can be seen at the end of this section.

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)


t_i OR t4
t1
t2
t3
t4
t_i OR t1
FCLK
CTTL
NTSO
RAS
CASH, L
DATA FROM RAM (READ)
READ DATA
DATA FROM CPU (WRITE)
ADDRESS
DATA TO BE WRITTEN

TL/L/9991-92

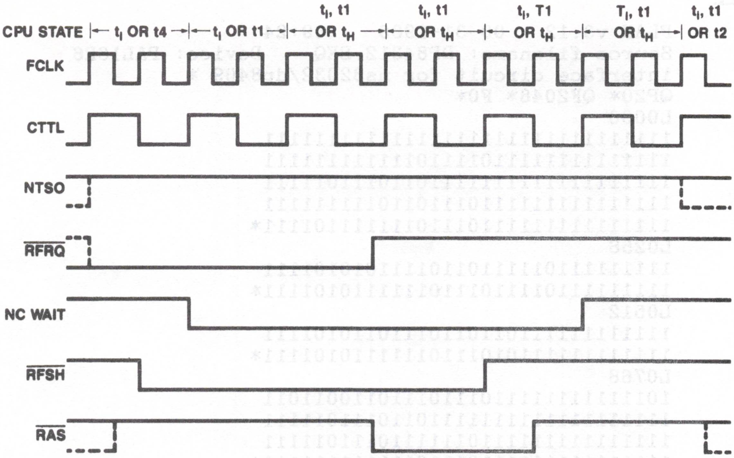
FIGURE 7.19.3. Timing Diagram; Read, Write or Hidden Refresh Memory Cycle for the NS32032-DP8409 Interface


CPU STATE
t4 OR t_i
t1
t2
tw
t3
t4
t1 OR t_i
FCLK
CTTL
NTSO
NC WAIT
RAS
CASH, L
DATA FROM RAM
VALID
DATA FROM CPU
ADDRESS
DATA TO BE WRITTEN TO MEMORY

TL/L/9991-93

FIGURE 7.19.4. Timing Diagram; Read, or Write Memory Cycle with One Wait

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)


CPU STATE
t_I OR t4
t_I OR t1
t_I, t1 OR t_H
t_I, t1 OR t_H
t_I, T1 OR t_H
T_I, t1 OR t_H
t_I, t1 OR t2
FCLK
CTTL
NTSO
RFRQ
NC WAIT
RFSH
RAS

TL/L/9991-94

FIGURE 7.19.5. Timing Diagram; Forced Refresh Cycle

PLAN™ INPUT FILE

```
PAL16R6; interface circuit for ns32032/dp8409
ck ntso /rfrq /hbe a0 /waitin cttl /cs /slow gnd
/oe /wait /d /c /b /a /casl /cash /rfsh vcc
/cash := /a* b*/c*/d*/hbe*/cs + a* b*/d*/hbe*/cs
/casl := /a* b* c*/d* a0*/cs +  a* b*/d* a0*/cs
/a :=  a* b* c* d*/ntso*/cs*/slow
   + /b* c* d* + /a* c* d + /a*/b
/b :=  a* b* c* d* ntso*/rfrq* cttl
   +  a*/b + /a*/b* c + /b*/c*/d
/c :=  a* b* c* d* ntso*/rfrq* cttl
   +  a* b*/d + /a*/b*/d + /b*/c* d +  a* b*/c* d*/ntso
/d :=  a* b* c* d*/ntso*/cs* slow
   +  a* b* c* d*/ntso* cs + /a* c +  b* c*/d +  a*/b*/c
/wait =  b* c* d*/ntso*/cs*/slow
     +  a*/b*/d +  a*/b* c + /b* c* d + /a*/b + /a*/c* d +  cs*/waitin
/rfsh =  a*/b + /b* c* d + /a*/b* c + /a*/b*/d
```

TL/L/9991-G7

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)

PLAN™ JEDEC FILE

```
PLAN v3.12  08-31-1988     09:24
Source filename: DP84312.BEQ    Device: PAL16R6
interface circuit for ns32032/dp8409 *
QP20* QF2048* F0*
L0000
11111111111111111111111111111111
11111111111111011110111111111111
11111111111111111110110111011111
11111111111111101110110111111111
11111111111111101110111111101111*
L0256
11111111101111101101111010101111
11111111101111011101111110101111*
L0512
11111111111101101101110110101111
11111111111101011101111110101111*
L0768
10111111111111011101110110011011
11111111111111111110110111011111
11111111111111101111110111011111
11111111111111101110111111111111*
L1024
01111011111111011101010111011111
11111111111111011110111111111111
11111111111111101110110111111111
11111111111111111110111011101111*
L1280
01111011111111011101010111011111
11111111111111011101111111101111
11111111111111101110111111101111
11111111111111111110111011011111
10111111111111011101111011011111*
L1536
10111111111111011101110110010111
10111111111111011101110101011111
11111111111111101111110111111111
11111111111111111101110111101111
11111111111111011110111011111111*
L1792
11111111111111111111111111111111
10111111111111111101110110011011
11111111111111011110111111101111
11111111111111011110110111111111
11111111111111111110110111011111
11111111111111101110111111111111
11111111111111101111111011011111
11111111111111111011111101111111*
C794A* 0B2F
```

TL/L/9991-G8

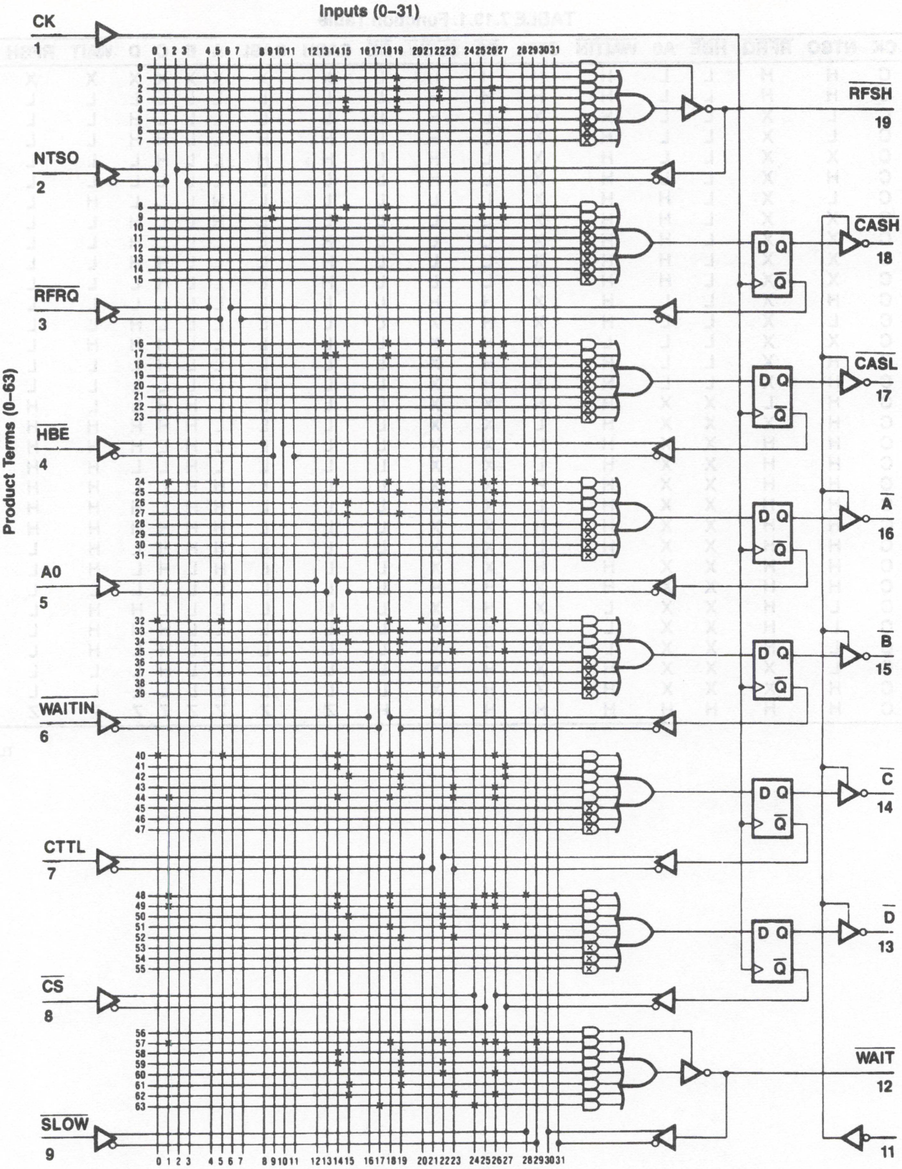
7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)

TABLE 7.19.1. Function Table

CK	NTSO	$\overline{\text{RFRQ}}$	$\overline{\text{HBE}}$	A0	$\overline{\text{WAITIN}}$	CTTL	$\overline{\text{CS}}$	$\overline{\text{SLOW}}$	$\overline{\text{OE}}$	CASH	CASL	A	B	C	D	WAIT	RFSH
C	H	H	L	L	H	H	H	H	L	X	X	X	X	X	X	X	X
C	H	H	L	L	H	H	H	H	L	L	L	L	L	L	L	L	L
C	L	X	L	L	H	X	L	H	L	L	L	L	L	L	H	L	L
C	L	X	L	L	H	X	L	H	L	H	H	L	L	H	H	L	L
C	X	X	L	L	H	X	L	H	L	H	H	L	L	H	L	L	L
C	H	X	L	L	H	X	L	H	L	L	L	L	L	L	L	L	L
C	L	X	L	H	H	X	L	L	L	L	L	H	L	L	L	H	L
C	X	X	L	H	H	X	L	L	L	L	L	H	L	L	H	L	L
C	X	X	L	H	H	X	L	L	L	H	L	L	L	L	H	L	L
C	X	X	L	H	H	X	L	L	L	H	L	L	L	H	H	L	L
C	X	X	L	H	H	X	L	L	L	H	L	L	L	H	L	L	L
C	H	X	L	L	H	X	H	H	L	L	L	L	L	L	L	L	L
C	L	X	L	L	H	X	H	X	L	L	L	L	L	L	H	L	L
C	X	X	L	L	L	X	H	X	L	L	L	L	L	H	H	H	L
C	H	X	L	L	H	X	H	X	L	L	L	L	L	H	L	L	L
C	H	X	L	L	H	X	H	X	L	L	L	L	L	L	L	L	L
C	H	L	X	X	H	H	X	X	L	L	L	L	H	H	L	L	H
C	H	X	X	X	H	L	X	X	L	L	L	L	H	H	H	H	H
C	H	H	X	X	H	H	X	X	L	L	L	L	H	L	H	H	H
C	H	H	X	X	H	L	X	X	L	L	L	L	H	L	L	H	H
C	H	H	X	X	H	H	X	X	L	L	L	H	H	L	L	H	H
C	H	H	X	X	H	L	X	X	L	L	L	H	H	L	H	H	H
C	H	H	X	X	H	H	X	X	L	L	L	H	H	H	H	H	H
C	H	H	X	X	H	L	X	X	L	L	L	H	H	H	L	H	L
C	H	H	X	X	H	H	X	X	L	L	L	H	L	H	L	H	L
C	H	H	X	X	H	X	H	H	L	L	L	L	L	L	L	L	L
C	L	H	X	X	L	X	H	X	L	L	L	L	L	L	H	H	L
C	L	H	X	X	L	X	H	X	L	L	L	L	L	H	H	H	L
C	L	H	X	X	L	X	H	X	L	L	L	L	L	H	L	H	L
C	L	X	X	X	H	X	H	X	L	L	L	L	L	H	L	L	L
C	H	X	X	X	H	X	H	X	L	L	L	L	L	L	L	L	L
C	H	H	H	H	H	H	H	H	H	Z	Z	Z	Z	Z	Z	Z	Z

TL/L/9991-97

7.19 DP84312 Dynamic RAM Controller Interface Circuit for the NS32032 CPU (Continued)


Inputs (0–31)
Product Terms (0–63)
CK
1
NTSO
2
RFRQ
3
HBE
4
A0
5
WAITIN
6
CTTL
7
CS
8
SLOW
9
RFSH
19
CASH
18
CASL
17
A
16
B
15
C
14
D
13
WAIT
12
11
D Q
Q

TL/L/9991–98

FIGURE 7.19.6. PAL16R6 Logic Diagram Showing DP84312 Pattern

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU*

GENERAL DESCRIPTION

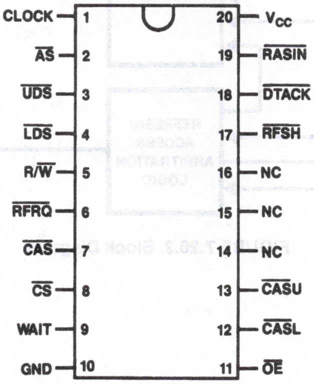
The DP84322 dynamic RAM controller interface is a PAL device for interface between the DP8409 dynamic RAM Controller and the 68000 microcprocessor.

The DP84322 supplies all the control signals needed to perform memory read, write and refresh. Logic is included for inserting a wait state when using fast CPUs.

FEATURES

- Provides 3-chip solution for the 68000 CPU and dynamic RAM interface
- Works with all 68000 speed versions
- Possibility of operation at 8 MHz with no wait states
- Performs hidden refresh
- DTACK is automatically inserted for both memory access and memory refresh
- Performs forced refresh using typically 4 CPU clocks
- Standard National Semiconductor PAL device part (PAL16R4)
- PAL device logic equations can be modified by the user for his specific application and programmed into any of National's PAL device family, including the new high speed PAL devices

Dual-In-Line Package



TL/L/9991-99

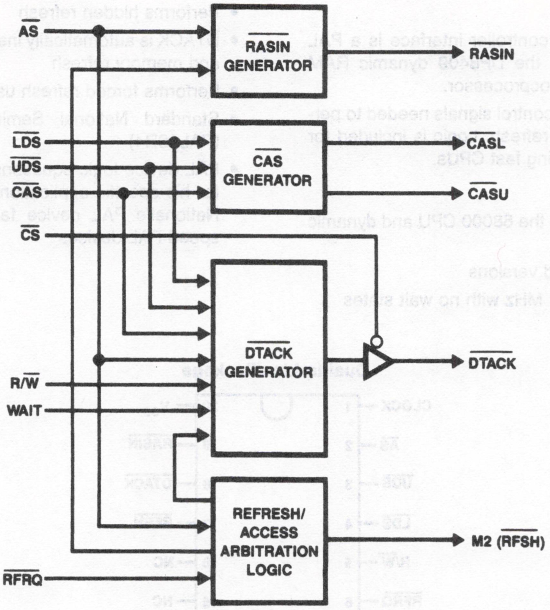
Top View

FIGURE 7.20.1. Connection Diagram

---

*Applications contained in this section are for illustration purposes only and National makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

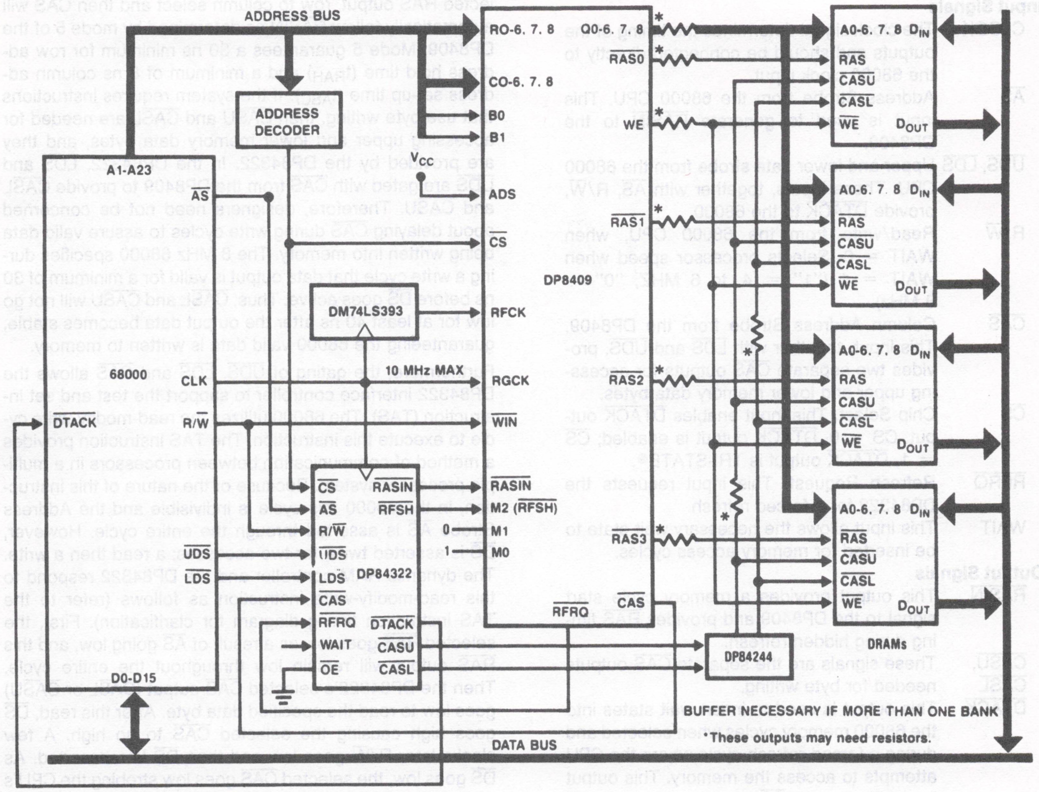

AS
LDS
UDS
CAS
CS
R/W
WAIT
RFRQ
RASIN GENERATOR
CAS GENERATOR
DTACK GENERATOR
REFRESH/ ACCESS ARBITRATION LOGIC
RASIN
CASL
CASU
DTACK
M2 (RFSH)

TL/L/9991-A0

FIGURE 7.20.2. Block Diagram

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

DP84322 and DP8409 for 68000 CPU


ADDRESS BUS
RO-6, 7, 8
CO-6, 7, 8
ADDRESS DECODER
B0
B1
VCC
A1-A23
AS
ADS
CS
DM74LS393
RFCK
68000
CLK
10 MHz MAX
RGCK
DTACK
R/W
WIN
CS
AS
R/W
UDS
LDS
CAS
RFRQ
WAIT
OE
RASIN
RFSH
0
DTACK
CASU
CASL
DP84322
RASIN
M2 (RFSH)
M1
M0
UDS
LDS
D0-D15
DATA BUS
DP8409
Q0-6, 7, 8
RAS0
WE
RAS1
RAS2
RAS3
CAS
RFRQ
DP84244
A0-6, 7, 8
DIN
RAS
CASU
CASL
WE
DOUT
DRAMs
*

BUFFER NECESSARY IF MORE THAN ONE BANK

*These outputs may need resistors.

TL/L/9991-A1

FIGURE 7.20.3. Synchronous Block Diagram

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

MNEMONIC DESCRIPTION

Input Signals

CLOCK	The clock signal determines the timing of the outputs and should be connected directly to the 68000 clock input.
$\overline{\text{AS}}$	Address Strobe from the 68000 CPU. This input is used to generate $\overline{\text{RASIN}}$ to the DP8409.
$\overline{\text{UDS}}$, $\overline{\text{LDS}}$	Upper and lower data strobe from the 68000 CPU. These inputs, together with $\overline{\text{AS}}$, R/$\overline{\text{W}}$, provide $\overline{\text{DTACK}}$ to the 68000.
R/$\overline{\text{W}}$	Read/write from the 68000 CPU, when WAIT = 0. Selects processor speed when WAIT = 1 ("1" = 4, to 6 MHZ, "0" = 8 MHz).
$\overline{\text{CAS}}$	Column Address Strobe from the DP8409. This input, together with $\overline{\text{LDS}}$ and $\overline{\text{UDS}}$, provides two separate $\overline{\text{CAS}}$ outputs for accessing upper and lower memory data bytes.
$\overline{\text{CS}}$	Chip Select. This input enables $\overline{\text{DTACK}}$ output. CS = 0, $\overline{\text{DTACK}}$ output is enabled; $\overline{\text{CS}}$ = 1, $\overline{\text{DTACK}}$ output is TRI-STATE®.
$\overline{\text{RFRQ}}$	Refresh Request. This input requests the DP84322 for a forced refresh.
WAIT	This input allows the necessary wait state to be inserted for memory access cycles.

Output Signals

$\overline{\text{RASIN}}$	This output provides a memory cycle start signal to the DP8409 and provides $\overline{\text{RAS}}$ timing during hidden refresh.
$\overline{\text{CASU}}$, $\overline{\text{CASL}}$	These signals are the separate $\overline{\text{CAS}}$ outputs needed for byte writing.
$\overline{\text{DTACK}}$	This output is used to insert wait states into the 68000 memory cycles when selected and during a forced refresh cycle where the CPU attempts to access the memory. This output is enabled when $\overline{\text{CS}}$ input is low and TRI-STATE when $\overline{\text{CS}}$ is high.
$\overline{\text{RFSH}}$	This output controls the mode of the DP8409. It always goes low for 4 CPU clock periods when $\overline{\text{AS}}$ is inactive and a forced refresh is requested through $\overline{\text{RFRQ}}$ input. This allows the DP8409 to perform an automatic forced refresh.

FUNCTIONAL DESCRIPTION

As a 68000 bus cycle begins, a valid address is output on the address bus A1–A23. This address is decoded to provide Chip Select ($\overline{\text{CS}}$) to the DP8409. After the address becomes valid, $\overline{\text{AS}}$ goes low and it is used to set $\overline{\text{RASIN}}$ low from the DP84322 interface circuit. Note that $\overline{\text{CS}}$ must go low for a minimum of 10 ns before the assertion of $\overline{\text{RASIN}}$ for a proper memory access. As an example, with an 8 MHz 68000, the address is valid for at least 30 ns before $\overline{\text{AS}}$ goes active. $\overline{\text{AS}}$ then has to ripple through the DP84322 to produce $\overline{\text{RASIN}}$. This means the address is valid for a minimum of 40 ns before $\overline{\text{RASIN}}$ goes low, and the decoding of $\overline{\text{CS}}$ should take less than 30 ns. At this speed the DM74LS138 or DM74LS139 decoders can be selected to guarantee the 10 ns minimum required by $\overline{\text{CS}}$ set-up time going low before the access $\overline{\text{RASIN}}$ goes low (t_{CSRL} of the DP8409). This is important because a false hidden refresh may take place when the minimum t_{CSRL} is not met.

Typically $\overline{\text{RASIN}}$ occurs at the end of S2. Subsequently, selected $\overline{\text{RAS}}$ output, row to column select and then $\overline{\text{CAS}}$ will automatically follow $\overline{\text{RASIN}}$ as determined by mode 5 of the DP8409. Mode 5 guarantees a 30 ns minimum for row address hold time (t_{RAH}) and a minimum of 8 ns column address set-up time (t_{ASC}). If the system requires instructions that use byte writing, then $\overline{\text{CASU}}$ and $\overline{\text{CASL}}$ are needed for accessing upper and lower memory data bytes, and they are provided by the DP84322. In the DP84322, $\overline{\text{LDS}}$ and $\overline{\text{UDS}}$ are gated with $\overline{\text{CAS}}$ from the DP8409 to provide $\overline{\text{CASL}}$ and $\overline{\text{CASU}}$. Therefore, designers need not be concerned about delaying $\overline{\text{CAS}}$ during write cycles to assure valid data being written into memory. The 8 MHz 68000 specifies during a write cycle that data output is valid for a minimum of 30 ns before $\overline{\text{DS}}$ goes active. Thus, $\overline{\text{CASL}}$ and $\overline{\text{CASU}}$ will not go low for at least 40 ns after the output data becomes stable, guaranteeing the 68000 valid data is written to memory.

Furthermore, the gating of $\overline{\text{UDS}}$, $\overline{\text{LDS}}$ and $\overline{\text{CAS}}$ allows the DP84322 interface controller to support the test and set instruction (TAS). The 68000 utilizes the read-modify-write cycle to execute this instruction. The TAS instruction provides a method of communication between processors in a multiple processor system. Because of the nature of this instruction, in the 68000 this cycle is indivisible and the Address Strobe $\overline{\text{AS}}$ is asserted through the entire cycle. However, $\overline{\text{DS}}$ is asserted twice for two accesses: a read then a write. The dynamic RAM controller and the DP84322 respond to this read-modify-write instruction as follows (refer to the TAS instruction timing diagram for clarification). First, the selected $\overline{\text{RAS}}$ goes low as a result of $\overline{\text{AS}}$ going low, and this $\overline{\text{RAS}}$ output will remain low throughout the entire cycle. Then the DP84322's selected $\overline{\text{CAS}}$ output ($\overline{\text{CASL}}$ or $\overline{\text{CASU}}$) goes low to read the specified data byte. After this read, $\overline{\text{DS}}$ goes high causing the selected $\overline{\text{CAS}}$ to go high. A few clocks later R/$\overline{\text{W}}$ goes low and then $\overline{\text{DS}}$ is reasserted. As $\overline{\text{DS}}$ goes low, the selected $\overline{\text{CAS}}$ goes low strobing the CPU's modified data into memory, after which the cycle is ended when $\overline{\text{AS}}$ goes high.

The two $\overline{\text{CAS}}$ outputs from the DP84322, however, can only drive one memory bank. For additional driving capability, a memory driver such as the DP84244 should be added to drive loads of up to 500 pF.

Since this DP84322 interface circuit is designed to operate with all of the 68000 speed versions, a status input called WAIT is used to distinguish the 8 MHz from the others. The WAIT input should be set low for a 6 MHz or less allowing full speed of operation with no wait states. Data Transfer Acknowledge input ($\overline{\text{DTACK}}$) of the 68000 at these speeds is automatically inserted during S2 for every memory transaction cycle and is then negated at the end of that cycle when $\overline{\text{UDS}}$ and/or $\overline{\text{LDS}}$ go high. For the 8 MHz 68000, however, a wait state is required for every memory transaction cycle. At these speeds, the WAIT input is set high, selecting the DP8409's $\overline{\text{CAS}}$ output to generate $\overline{\text{DTACK}}$ and again $\overline{\text{DTACK}}$ is negated at the end of the cycle when $\overline{\text{UDS}}$ or $\overline{\text{LDS}}$ goes high. Note that $\overline{\text{DTACK}}$ output is enabled only when the DP8409's $\overline{\text{CS}}$ is low. Therefore when the 68000 is accessing I/O or ROM (in other words, when the DP8409 is not selected), the DP84322's $\overline{\text{DTACK}}$ output goes high impedance logic '1' through the external pull-up resistor and it is now up to the designer to supply $\overline{\text{DTACK}}$ for a proper bus cycle.

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

Table 7.2.1 indicates the maximum memory speed in terms of the DRAM timing parameters: t_{CAC} (access-time from $\overline{\text{CAS}}$) and t_{RP} ($\overline{\text{RAS}}$ precharge time) required by different 68000 speed versions.

TABLE 7.20.1 Memory Speed

Microprocessor Clock	Maximum t_{CAC}	Minimum t_{RP}	Minimum t_{RAS}
8 MHz	125 ns	140 ns	220 ns
6 MHz	90 ns	170 ns	290 ns
4 MHz	270 ns	280 ns	450 ns

Pin 5 (R/$\overline{\text{W}}$ input to the DP84322) is not used as R/$\overline{\text{W}}$ when the WAIT input is high. Therefore, when WAIT is high and pin 5 is low, this is configured for the 8 MHz 68000. The dynamic RAM controller in this configuration operates in mode 5 and mode 1.

When both WAIT and pin 5 are high, this is configured for 4 MHz and 6 MHz 68000, allowing only two microprocessor clocks for memory refresh. Furthermore, the designer can use the DP8408 because the dynamic RAM controller now operates in mode 0 and mode 5 or mode 6. In addition, the programmable refresh timer, DP84300, should be used to determine the refresh rate (RFCK) and to provide the refresh request ($\overline{\text{RFRQ}}$) input to the DP84322. The refresh timer can provide over two hundred different divisors. $\overline{\text{RFRQ}}$ is given at the beginning of every RFCK cycle and remains active until M2 goes low for memory refresh. The DP84322 samples $\overline{\text{RFRQ}}$ when $\overline{\text{AS}}$ is high, then sets M2 low for two microprocessor clocks, taking the DP8408 or DP8409 to the external control refresh mode. $\overline{\text{RASIN}}$ for this refresh is also issued by the DP84322. If a memory access is pending, $\overline{\text{RASIN}}$ for this access will not be given until it is delayed for approximately one microprocessor clock, allowing $\overline{\text{RAS}}$ precharge time for the dynamic RAMs.

Table 7.20.2 indicates different memory speeds in terms of the DRAM parameters required by 4 MHz and 6 MHz 68000.

TABLE 7.20.2. Memory Speed of 68000

Microprocessor Clock	Maximum t_{CAC}	Minimum t_{RAS}	Minimum t_{RP}	Minimum t_{RAH}
4 MHz	290 ns	200 ns	225 ns	20 ns
6 MHz	110 ns	125 ns	140 ns	20 ns

When WAIT = 1, pin 5 = 0 (8 MHz), the PAL device controller supports read and write cycles with one inserted wait state, forced refresh with five wait states inserted if $\overline{\text{CS}}$ is valid, and hidden refresh. This PAL device mode does not support the TAS instruction.

When WAIT = pin 5 = 1 (4–6 MHz), the PAL device controller supports read and write cycles with no wait states inserted, and forced refresh with two wait states inserted if $\overline{\text{CS}}$ is valid. This PAL device mode does not support the TAS instruction and only supports hidden refresh when used in mode 5 with the DP8409 controller.

The DP84322 can possibly be operated at 8 MHz with no wait states (WAIT = "0") given the following conditions:

Fast PAL Device (PAL 16R4A)

S2 + S3 + S4 + S5 = 250 ns

$\overline{\text{RASIN}}$ delay = 60 ns ($\overline{\text{AS}}$ low max)
+25 ns (Fast PAL delay) = 85 ns max

$\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ delay DP8409-2 = 130 ns max

External $\overline{\text{CASH,L}}$ generation using 74S02 and 74S240
7.5 ns (74S02) + 10 ns (74S240) − 7.5 ns (less load on 8409 $\overline{\text{CAS}}$ line) = 10 ns max

Transceiver delay (74LS245) = 12 ns max

68000 data setup into S6 = 40 ns min

∴ Minimum t_{CAC} = 53 ns
= 250 − 85 − 130 − 10 − 12 + 40

Minimum t_{RAS} = 240 ns

Minimum t_{RP} = 150 ns

Minimum t_{RAH} = 20 ns

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

REFRESH CYCLE

Since the access sequence timing is automatically derived from RASIN in mode 5, R/C and CASIN are not used and now become Refresh Clock (RFCK) and RAS-generator clock (RGCK) respectively. The Refresh Clock RFCK may be divided down from RGCK, which is the microprocessor clock, using the DM74LS393 or DM74LS390. RFCK provides the refresh time interval and RGCK the fast clock for all-RAS refresh if forced refreshing is necessary. The DP8409 offers both hidden refresh in mode 5 and forced refresh in mode 1 with priority placed on hidden refreshing. Assume 128 rows are being refreshed, then a 16 μs maximum clock period is needed for RFCK to distribute refreshing of all the rows over the 2 ms period.

The DP8409 provides hidden refreshing in mode 5 when the refresh clock (RFCK) is high and the microprocessor is accessing RAM. In other words, when the DP8409's chip select is inactive because the microprocessor is not accessing elsewhere, all four RAS outputs follow RASIN, strobing the contents of the on-chip refresh counter to every memory bank. RASIN going high terminates the hidden refresh and also increments the refresh counter, preparing it for the next refresh cycle. Once a hidden refresh has taken place, a forced refresh will not be requested by the DP8409 for the current RFCK cycle.

However, if the microprocessor continuously accessed the DP8409 and memory while RFCK was high, a hidden refresh could not have taken place and now the system must force a refresh. Immediately after RFCK goes low, the Refresh Request signal (RFRQ) from the DP8409 goes low, indicating a forced refresh is necessary. First, when RFRQ goes low any time during S2 to S7, the controller interface circuit waits until the end of the current memory access cycle and then sets M2 (RFSH) low. This refresh takes four microprocessor clocks to complete. If the current cycle is another memory cycle, the 68000 will automatically be put in four wait states.

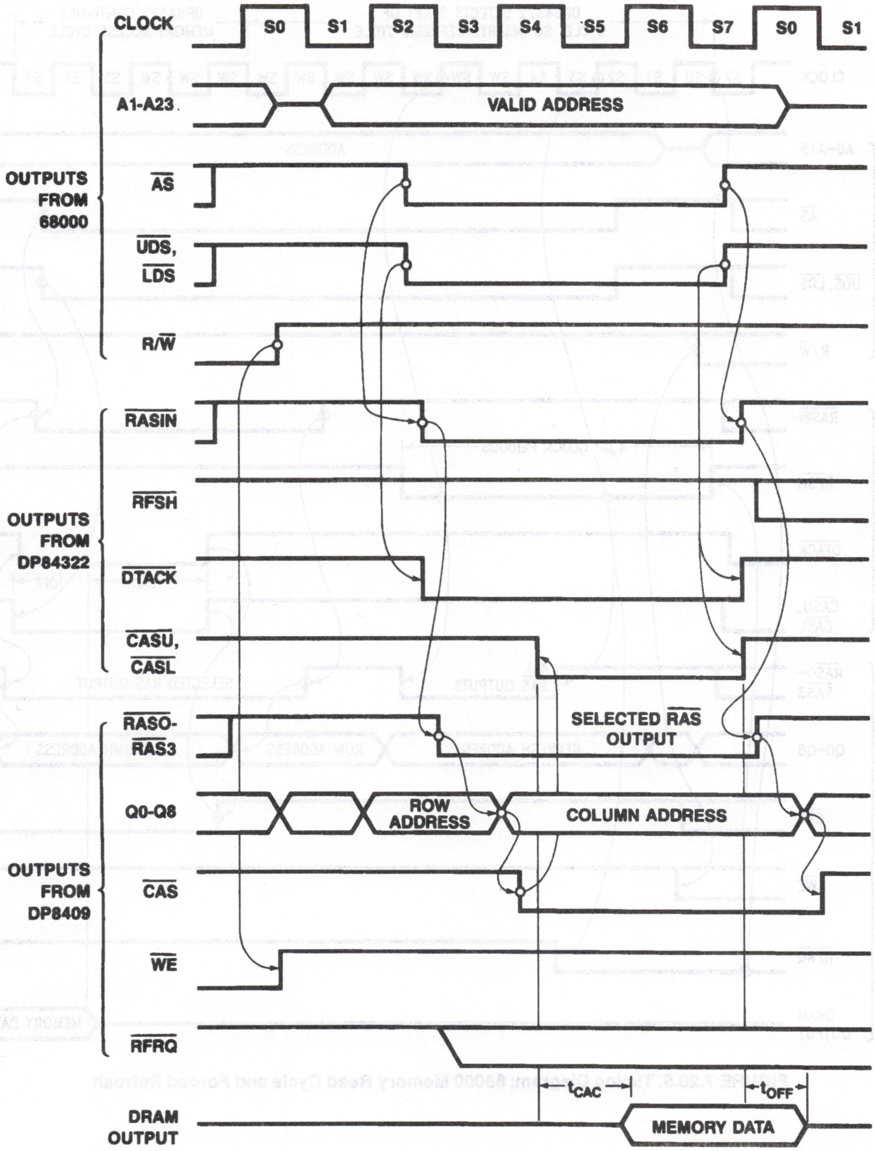
Alternately, when RFRQ goes low while AS is high during S0 to S1, M2 is now set low at S2. Therefore, it requires an additional microprocessor clock for this refresh. Once the DP8409 is in mode 1 forced refresh, all the RAS outputs remain high until two RGCK trailing edges after M2 goes low, when all RAS outputs go low. This allows a minimum of one and a half clock periods of RGCK for RAS precharge time. As specified in the DP8409 data sheet, the RAS outputs remain low for two clock periods of RGCK. The refresh counter is incremented as the RAS outputs go high. Once the forced refresh has ended, M2 is brought high, the DP8409 back to mode 5 auto access. Note that RASIN for the pending access is not given until it has been delayed for a full microprocessor clock, allowing RAS precharge time for the coming access.

If the 68000 bus is inactive (i.e., the 68000's instruction queue is full, or the 68000 is executing internal operations such as a multiply instruction, or the 68000 is in half state, etc.) and a refresh has been requested, a refresh will also take place because RFRQ is continuously sampled while AS is high. Therefore, refreshing under these conditions will be transparent to the microprocessor. Consequently, the system throughput is increased because the DP84322 allows refreshing while the 68000 bus is inactive.

The 84322 is a standard National PAL device part (PAL16R4). The user can modify the PAL equations to support his particular application. The 84322 logic equations, function table, and logic diagram can be seen at the end of this section.

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

68000 Memory Read Cycle (WAIT = 0, Pin 5 = R/W)

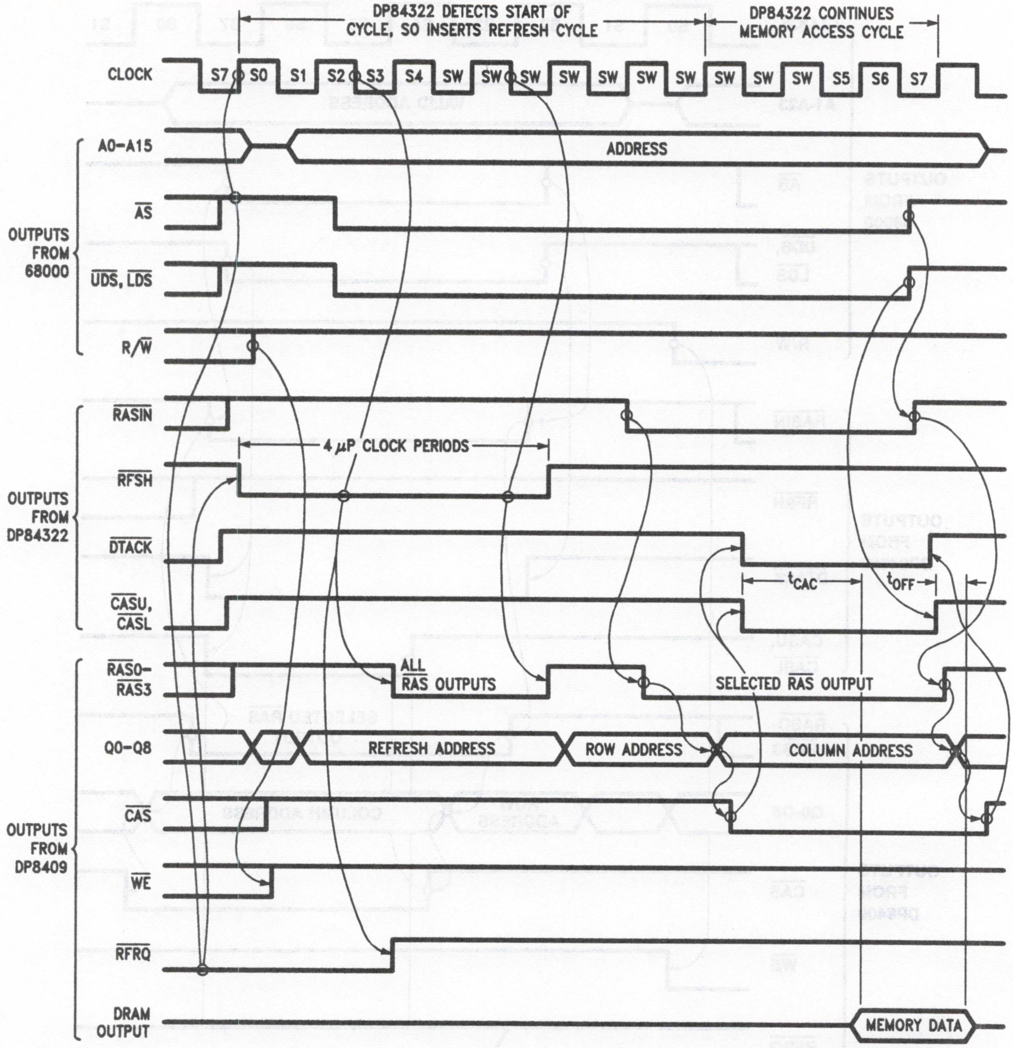

CLOCK
S0
S1
S2
S3
S4
S5
S6
S7
S0
S1
A1-A23
VALID ADDRESS
OUTPUTS FROM 68000
AS
UDS, LDS
R/W
RASIN
RFSH
OUTPUTS FROM DP84322
DTACK
CASU, CASL
RAS0-RAS3
SELECTED RAS OUTPUT
Q0-Q8
ROW ADDRESS
COLUMN ADDRESS
OUTPUTS FROM DP8409
CAS
WE
RFRQ
t_{CAC}
t_{OFF}
DRAM OUTPUT
MEMORY DATA

TL/L/9991-A2

FIGURE 7.20.4. Timing Diagram; 68000 Memory Read Cycle

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

Memory Read Cycle and Forced Refresh (Wait = 1, Pin 5 = 0)

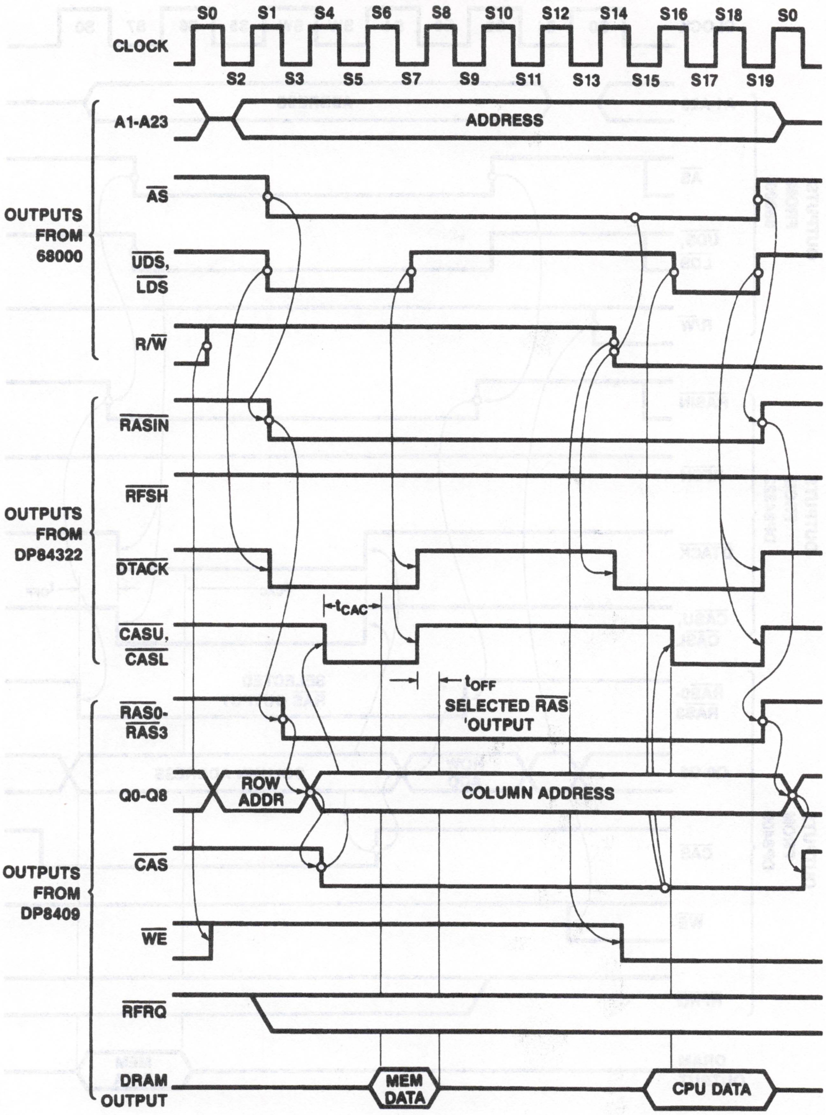

DP84322 DETECTS START OF CYCLE, SO INSERTS REFRESH CYCLE
DP84322 CONTINUES MEMORY ACCESS CYCLE
CLOCK
S7 S0 S1 S2 S3 S4 SW SW SW SW SW SW SW SW SW SW S5 S6 S7
OUTPUTS FROM 68000
A0–A15
ADDRESS
$\overline{AS}$
$\overline{UDS}$, $\overline{LDS}$
R/$\overline{W}$
OUTPUTS FROM DP84322
$\overline{RASIN}$
4 µP CLOCK PERIODS
$\overline{RFSH}$
$\overline{DTACK}$
t_{CAC}
t_{OFF}
$\overline{CASU}$, $\overline{CASL}$
OUTPUTS FROM DP8409
$\overline{RAS0}$–$\overline{RAS3}$
ALL $\overline{RAS}$ OUTPUTS
SELECTED $\overline{RAS}$ OUTPUT
Q0–Q8
REFRESH ADDRESS
ROW ADDRESS
COLUMN ADDRESS
$\overline{CAS}$
$\overline{WE}$
$\overline{RFRQ}$
DRAM OUTPUT
MEMORY DATA

TL/L/9991–A3

FIGURE 7.20.5. Timing Diagram; 68000 Memory Read Cycle and Forced Refresh

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

TAS Instruction Cycle (WAIT = 0, Pin 5 = R/$\overline{W}$)

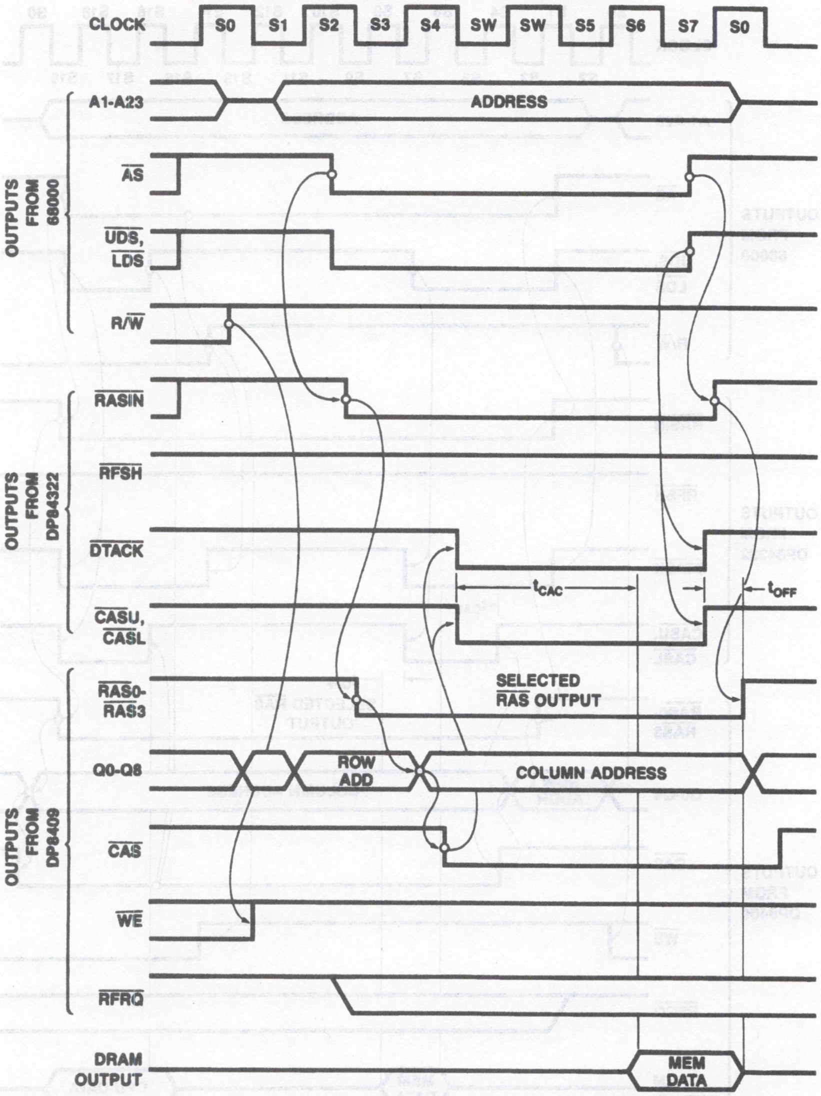

S0
S1
S4
S6
S8
S10
S12
S14
S16
S18
S0
CLOCK
S2
S3
S5
S7
S9
S11
S13
S15
S17
S19
A1-A23
ADDRESS
OUTPUTS FROM 68000
$\overline{AS}$
$\overline{UDS}$, $\overline{LDS}$
R/$\overline{W}$
OUTPUTS FROM DP84322
$\overline{RASIN}$
$\overline{RFSH}$
$\overline{DTACK}$
$\overline{CASU}$, $\overline{CASL}$
t_{CAC}
t_{OFF}
SELECTED $\overline{RAS}$ OUTPUT
OUTPUTS FROM DP8409
$\overline{RAS0}$-$\overline{RAS3}$
Q0-Q8
ROW ADDR
COLUMN ADDRESS
$\overline{CAS}$
$\overline{WE}$
$\overline{RFRQ}$
DRAM OUTPUT
MEM DATA
CPU DATA

TL/L/9991-A4

FIGURE 7.20.6. Timing Diagram; TAS Instruction Cycle

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

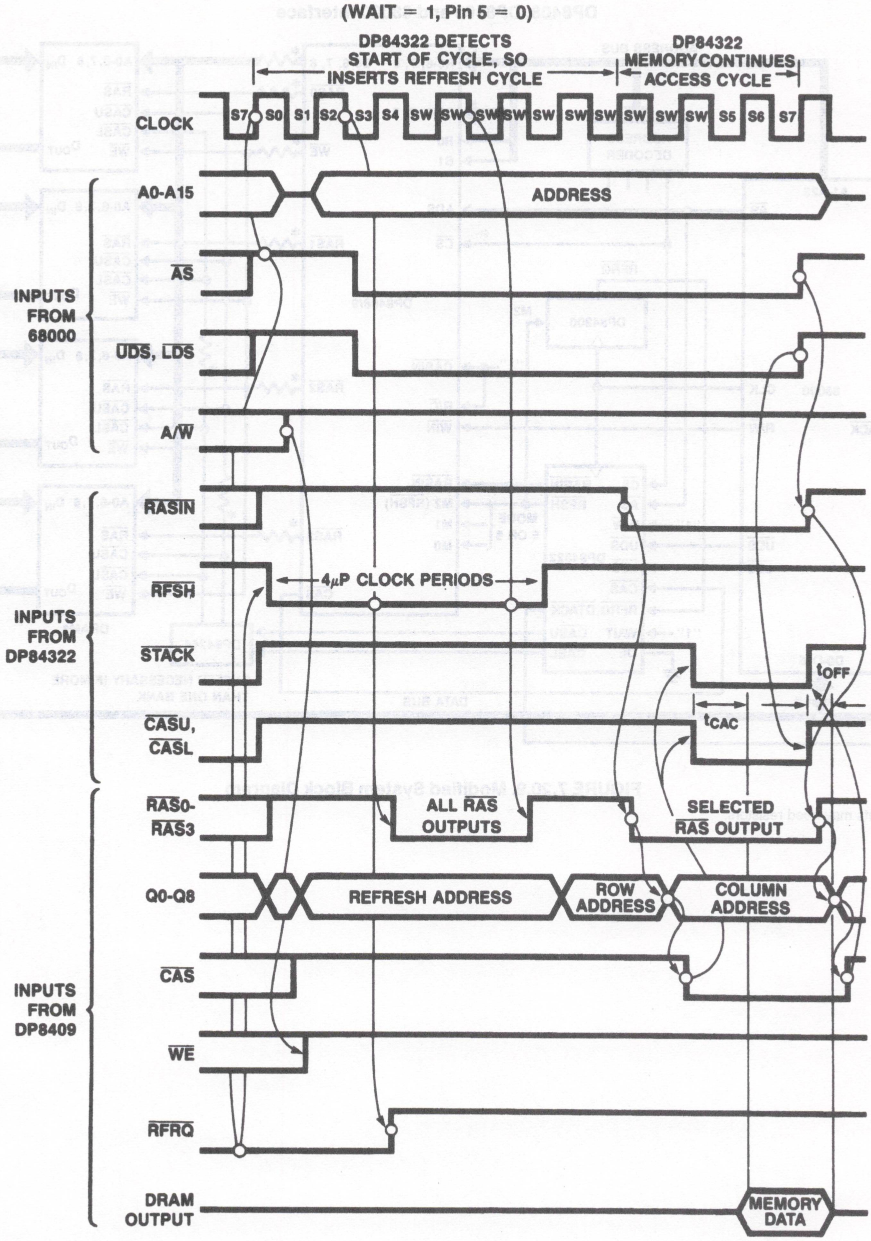
Memory Read Cycle (WAIT = 1, Pin 5 = 0)


CLOCK
S0
S1
S2
S3
S4
SW
SW
S5
S6
S7
S0
A1-A23
ADDRESS
AS
UDS,
LDS
R/W
OUTPUTS FROM 68000
RASIN
RFSH
DTACK
CASU,
CASL
OUTPUTS FROM DP84322
tCAC
tOFF
SELECTED RAS OUTPUT
RAS0-RAS3
Q0-Q8
ROW ADD
COLUMN ADDRESS
CAS
WE
RFRQ
OUTPUTS FROM DP8409
DRAM OUTPUT
MEM DATA

TL/L/9991-A5

FIGURE 7.20.7. Timing Diagram; Memory Read Cycle

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

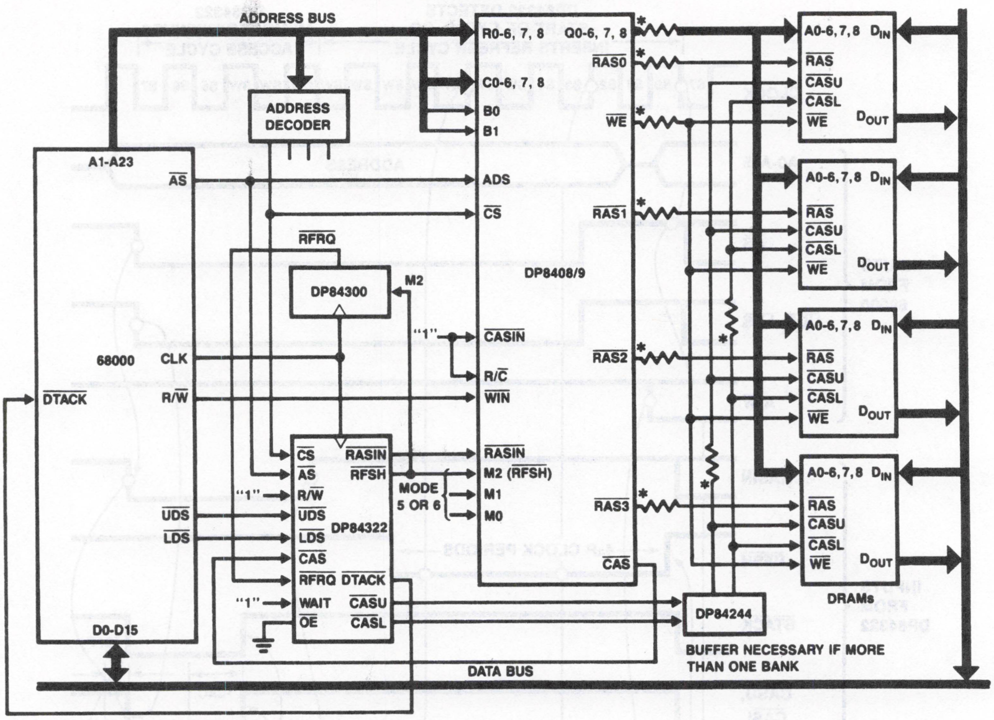

(WAIT = 1, Pin 5 = 0)
DP84322 DETECTS START OF CYCLE, SO INSERTS REFRESH CYCLE
DP84322 MEMORY CONTINUES ACCESS CYCLE
CLOCK
S7
S0
S1
S2
S3
S4
SW
S5
S6
A0-A15
ADDRESS
AS
UDS, LDS
R/W
INPUTS FROM 68000
RASIN
RFSH
4μP CLOCK PERIODS
STACK
CASU, CASL
tCAC
tOFF
INPUTS FROM DP84322
RAS0-RAS3
ALL RAS OUTPUTS
SELECTED RAS OUTPUT
Q0-Q8
REFRESH ADDRESS
ROW ADDRESS
COLUMN ADDRESS
CAS
WE
RFRQ
DRAM OUTPUT
MEMORY DATA
INPUTS FROM DP8409

TL/L/9991-A6

FIGURE 7.20.8. Timing Diagram; Memory Read Cycle and Forced Refresh

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

DP8408, DP8409 and 68000 Interface


ADDRESS BUS
ADDRESS DECODER
A1-A23
AS
68000
CLK
R/W
DTACK
UDS
LDS
D0-D15
RFRQ
DP84300
M2
"1"
CS
AS
R/W
UDS
LDS
CAS
RFRQ
WAIT
OE
RASIN
RFSH
DP84322
DTACK
CASU
CASL
MODE
5 OR 6
R0-6, 7, 8
C0-6, 7, 8
B0
B1
ADS
CS
DP8408/9
CASIN
R/C
WIN
RASIN
M2 (RFSH)
M1
M0
Q0-6, 7, 8
RAS0
WE
RAS1
RAS2
RAS3
CAS
*
A0-6, 7, 8
DIN
RAS
CASU
CASL
WE
DOUT
DRAMs
DP84244
BUFFER NECESSARY IF MORE THAN ONE BANK
DATA BUS

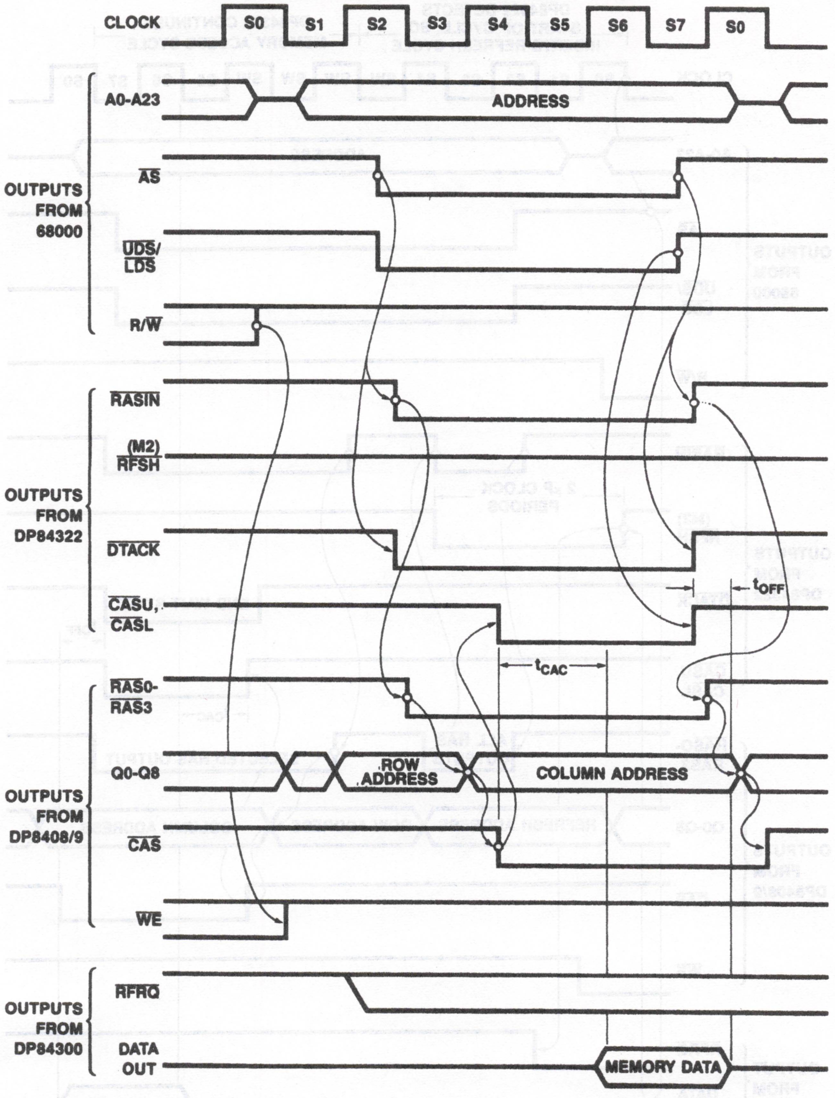
TL/L/9991-A7

FIGURE 7.20.9. Modified System Block Diagram

*These outputs may need resistors.

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

68000 Memory Read Cycle (WAIT and Pin 5 = 1)

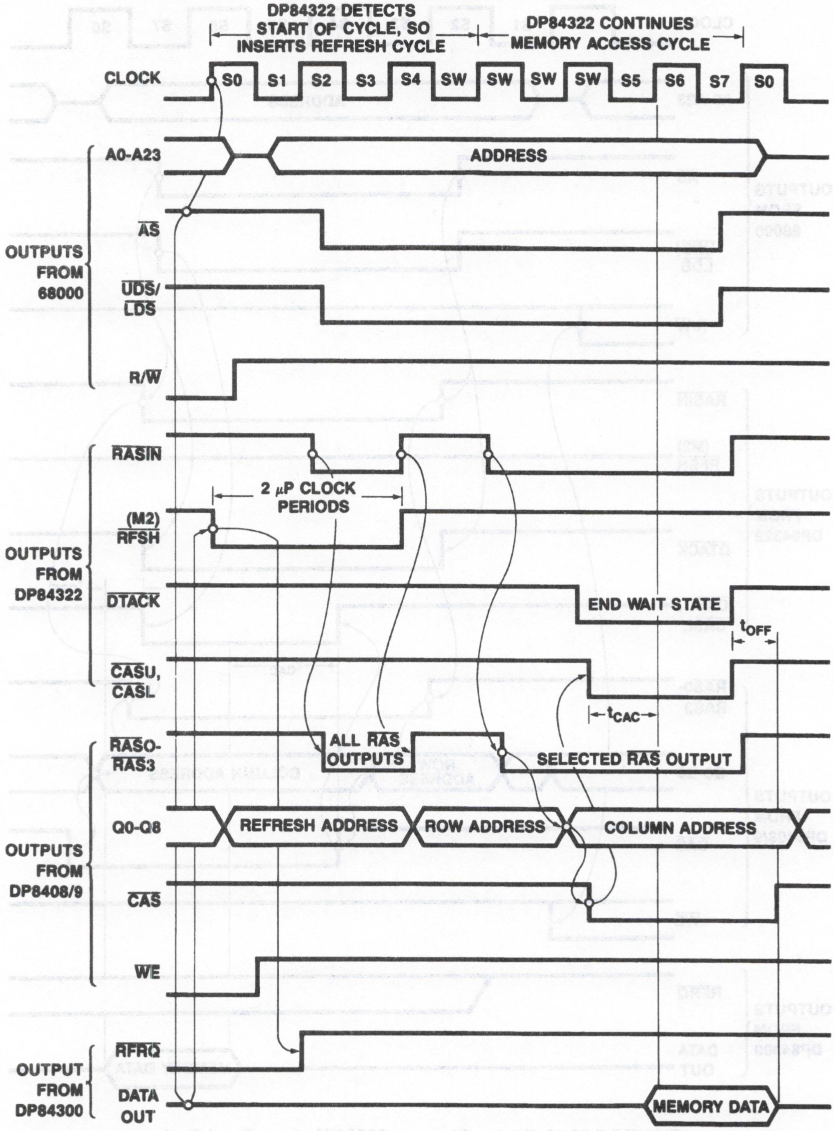

CLOCK
S0
S1
S2
S3
S4
S5
S6
S7
S0
OUTPUTS FROM 68000
A0-A23
ADDRESS
AS
UDS/LDS
R/W
OUTPUTS FROM DP84322
RASIN
(M2) RFSH
DTACK
CASU, CASL
tOFF
tCAC
OUTPUTS FROM DP8408/9
RAS0-RAS3
Q0-Q8
ROW ADDRESS
COLUMN ADDRESS
CAS
WE
OUTPUTS FROM DP84300
RFRQ
DATA OUT
MEMORY DATA

TL/L/9991-A8

FIGURE 7.20.10. Timing Diagram; 68000 Memory Read Cycle

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

68000 Memory Read Cycle and Memory Refresh (WAIT and Pin 5 = 0)


DP84322 DETECTS START OF CYCLE, SO INSERTS REFRESH CYCLE
DP84322 CONTINUES MEMORY ACCESS CYCLE
CLOCK
S0
S1
S2
S3
S4
SW
SW
SW
SW
S5
S6
S7
S0
A0-A23
ADDRESS
$\overline{AS}$
OUTPUTS FROM 68000
$\overline{UDS}$/$\overline{LDS}$
R/$\overline{W}$
$\overline{RASIN}$
2 µP CLOCK PERIODS
(M2) $\overline{RFSH}$
OUTPUTS FROM DP84322
DTACK
END WAIT STATE
t_{OFF}
$\overline{CASU}$, $\overline{CASL}$
t_{CAC}
$\overline{RAS0}$-$\overline{RAS3}$
ALL $\overline{RAS}$ OUTPUTS
SELECTED $\overline{RAS}$ OUTPUT
Q0-Q8
REFRESH ADDRESS
ROW ADDRESS
COLUMN ADDRESS
OUTPUTS FROM DP8408/9
$\overline{CAS}$
WE
$\overline{RFRQ}$
OUTPUT FROM DP84300
DATA OUT
MEMORY DATA

TL/L/9991-A9

FIGURE 7.20.11. Timing Diagram; 68000 Memory Read Cycle and Memory Refresh

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

PLAN™ INPUT FILE

```
PAL16R4; dram controller interface for mc68000/dp8409
ck /as /uds /lds r /rfrq /cas /cs wait gne
/oe /cl /cu /c /b /a /rfsh /dtack /rasin vcc
/rasin = /as* rfsh* a + /rfsh* r*/a* wait
/dtack = /r*/cas* wait + /uds* a* b*/wait + /lds* a* b*/wait
         + /as*/r* a* b*/wait + /as* rfsh* r* a* b* wait
dtack.trst = /cs
/rfsh :=  as*/rfrq + /rfsh*/r* c* wait
         + /rfsh* r* a* wait + /rfsh* c*/wait
/a := /rfsh
/b := /a
/c := /b
/cu = /uds*/cas
/cl = /lds*/cas
```

TL/L/9991-G9

PLAN™ JEDEC FILE

```
PLAN v3.12   09-02-1988    14:13
Source filename: DP84322.BEQ    Device: PAL16R4
dram controller interface for mc68000/dp8409 *
QP20* QF2048* F0*
L0000
11111111111111111111111111111111
10111111110111011111111111111111
11111111111001101111111111110111*
L0256
11111111111111111111111110111111
11111111111110111111101111110111
11111011111111011101111111111011
11111111101111011101111111111011
10111111111110011101111111111011
10111111110101011101111111110111*
L0512
01111111111111111011111111111111
11111111111010111111110111110111
11111111111001011111111111110111
11111111111011111111110111111011*
L0768
11111111111011111111111111111111*
L1024
11111111111111101111111111111111*
L1280
11111111111111111110111111111111*
L1536
11111111111111111111111111111111
11111011111111111111101111111111*
L1792
11111111111111111111111111111111
11111111101111111111101111111111*
C4A59* B141
```

TL/L/9991-H0

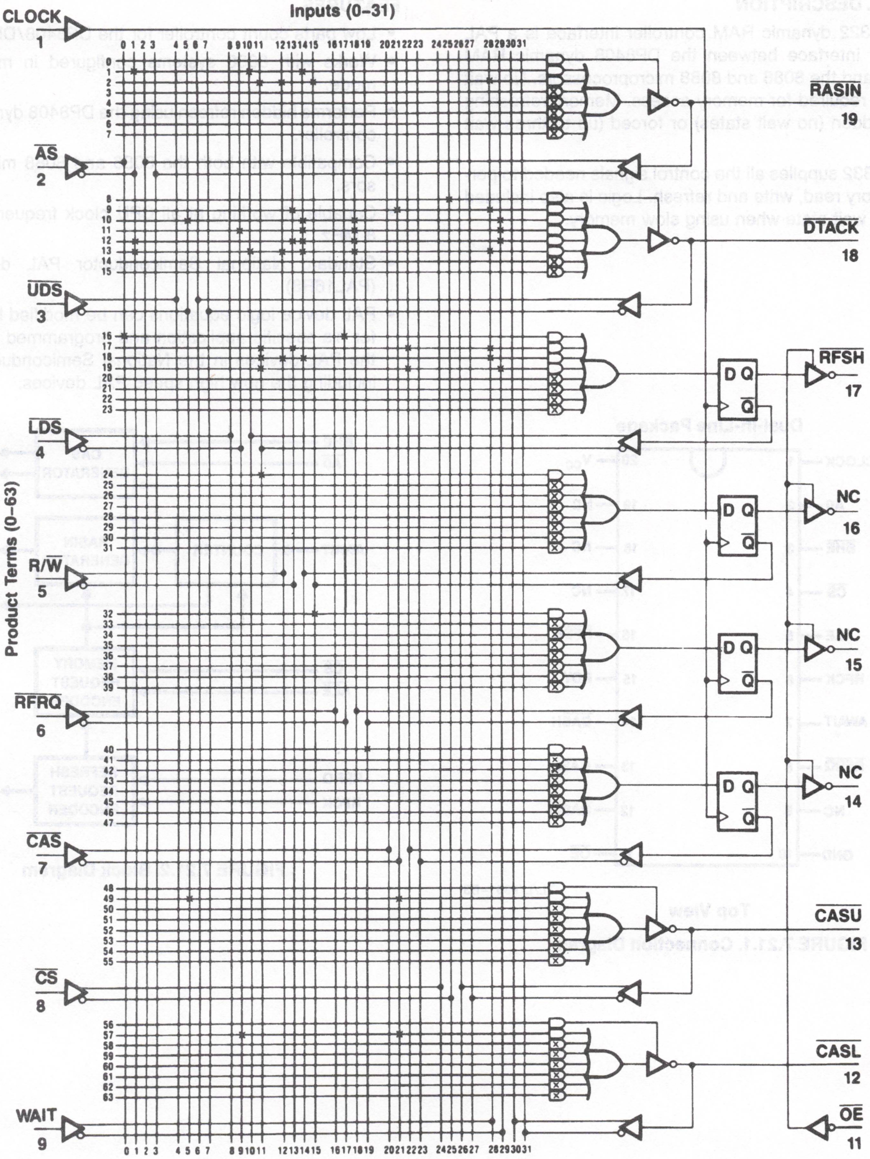
7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)

TABLE 7.20.3. Function Table

CK	$\overline{AS}$	$\overline{UDS}$	$\overline{LDS}$	R	$\overline{RFRQ}$	$\overline{CAS}$	$\overline{CS}$	WAIT	$\overline{OE}$	$\overline{CL}$	$\overline{CU}$	$\overline{C}$	$\overline{B}$	$\overline{A}$	RFSH	$\overline{DTACK}$	$\overline{RASIN}$
C	H	L	L	H	H	H	H	L	L	H	H	X	X	X	X	X	H
C	H	L	L	H	H	L	H	L	L	L	L	X	X	X	X	X	H
C	H	L	H	H	H	L	H	L	L	H	L	X	X	X	X	X	H
C	H	H	L	H	H	L	H	L	L	L	H	X	X	X	X	X	H
C	H	H	H	H	H	H	H	L	L	H	H	H	H	H	H	Z	H
C	L	L	H	H	H	H	L	L	L	H	H	H	H	H	H	L	L
C	L	L	H	H	H	L	L	L	L	H	L	H	H	H	H	L	L
C	L	H	H	H	H	L	L	L	L	H	H	H	H	H	H	H	L
C	L	H	H	L	H	L	L	L	L	H	H	H	H	H	H	L	L
C	L	L	H	L	H	L	L	L	L	H	L	H	H	H	H	L	L
C	H	H	H	L	H	H	L	L	L	H	H	H	H	H	H	H	H
C	H	H	H	L	L	H	L	L	L	H	H	H	H	H	L	H	H
C	H	H	H	L	L	H	L	L	L	H	H	H	H	L	L	H	H
C	L	H	L	L	H	H	L	L	L	H	H	H	L	L	L	H	H
C	L	H	L	L	H	H	L	L	L	H	H	L	L	L	L	H	H
C	L	H	L	L	H	H	L	L	L	H	H	L	L	L	H	H	H
C	L	H	L	L	H	H	L	L	L	H	H	L	L	H	H	H	L
C	L	H	L	L	H	L	L	L	L	L	H	L	H	H	H	L	L
C	L	H	L	L	H	L	L	L	L	L	H	H	H	H	H	L	L
C	H	H	H	L	H	L	L	L	L	H	H	H	H	H	H	H	H
C	H	H	H	L	L	H	L	H	L	H	H	H	H	H	L	H	H
C	H	H	H	L	L	H	L	H	L	H	H	H	H	L	L	H	H
C	L	L	L	L	H	H	L	H	L	H	H	H	L	L	L	H	H
C	L	L	L	L	H	H	L	H	L	H	H	L	L	L	L	H	H
C	L	L	L	L	H	H	L	H	L	H	H	L	L	L	H	H	H
C	L	L	L	L	H	H	L	H	L	H	H	L	L	H	H	H	L
C	L	L	L	L	H	L	L	H	L	L	L	L	H	H	H	L	L
C	H	H	H	L	H	L	L	H	L	H	H	H	H	H	H	L	H
C	H	H	H	L	H	H	H	H	L	H	H	H	H	H	H	Z	H
C	H	H	H	H	L	H	L	H	L	H	H	H	H	H	L	H	H
C	H	H	H	H	L	H	L	H	L	H	H	H	H	L	L	H	L
C	L	L	H	H	H	H	L	H	L	H	H	H	L	L	H	H	H
C	L	L	H	H	H	H	L	H	L	H	H	L	L	H	H	H	L
C	L	L	H	H	H	L	L	H	L	H	L	L	H	H	H	L	L
C	H	H	H	H	H	L	L	H	L	H	H	H	H	H	H	H	H
C	H	H	H	H	H	H	L	H	H	H	H	Z	Z	Z	Z	H	H

TL/L/9991-B1

7.20 DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU (Continued)


CLOCK
1
Inputs (0–31)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
$\overline{RASIN}$
19
$\overline{AS}$
2
$\overline{DTACK}$
18
$\overline{UDS}$
3
$\overline{RFSH}$
17
D Q
$\overline{Q}$
$\overline{LDS}$
4
NC
16
Product Terms (0–63)
R/W
5
NC
15
$\overline{RFRQ}$
6
NC
14
$\overline{CAS}$
7
$\overline{CASU}$
13
$\overline{CS}$
8
$\overline{CASL}$
12
WAIT
9
$\overline{OE}$
11

TL/L/9991–B2

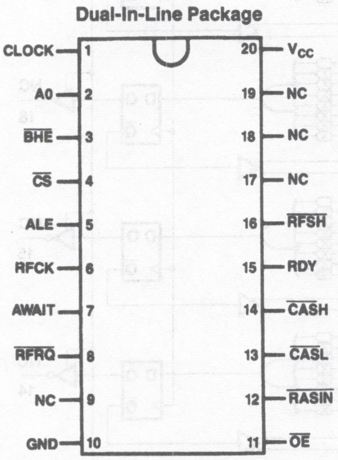
FIGURE 7.20.12. PAL16R4 Logic Diagram Showing DP84322 Pattern

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs*

GENERAL DESCRIPTION

The DP84322 dynamic RAM controller interface is a PAL device for interface between the DP8408 dynamic RAM controller and the 8086 and 8088 microprocessors. No wait states are required for memory access. Memory refreshing may be hidden (no wait states) or forced (up to three wait states).

The DP84332 supplies all the control signals needed to perform memory read, write and refresh. Logic is also included to insert a wait state when using slow memory.


Dual-In-Line Package
CLOCK 1
A0 2
$\overline{BHE}$ 3
$\overline{CS}$ 4
ALE 5
RFCK 6
AWAIT 7
$\overline{RFRQ}$ 8
NC 9
GND 10
20 V_{CC}
19 NC
18 NC
17 NC
16 $\overline{RFSH}$
15 RDY
14 $\overline{CASH}$
13 $\overline{CASL}$
12 $\overline{RASIN}$
11 $\overline{OE}$

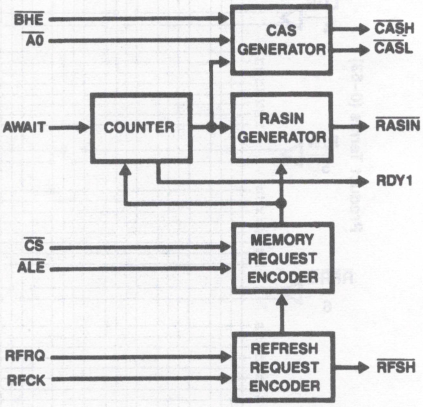
TL/L/9991-B3

Top View

FIGURE 7.21.1. Connection Diagram

FEATURES

- Low parts count controller for the DP8408/DP8409.
- Works with 8086 systems configured in min or max mode.
- Performs hidden refresh using the DP8408 dynamic RAM controller.
- Compatible with both the 8086 and 8088 microprocessors.
- Capable of working at all CPU clock frequencies up to 8 MHz.
- Standard National Semiconductor PAL device part (PAL16R8).
- PAL device logic equations can be modified by the user for his specific application and programmed into any of the PAL devices in the National Semiconductor family, including the new high speed PAL devices.


$\overline{BHE}$
$\overline{A0}$
CAS GENERATOR
$\overline{CASH}$
$\overline{CASL}$
AWAIT
COUNTER
RASIN GENERATOR
$\overline{RASIN}$
RDY1
$\overline{CS}$
$\overline{ALE}$
MEMORY REQUEST ENCODER
RFRQ
RFCK
REFRESH REQUEST ENCODER
$\overline{RFSH}$

TL/L/9991-B4

FIGURE 7.21.2. Block Diagram

---

*Applications contained in this section are for illustration purposes only and National makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)

MNEMONIC DESCRIPTION

Input Signals

CLOCK	The clock signal determines the timing of the outputs and should be connected directly to the 8086 clock.
A0, $\overline{\text{BHE}}$	These inputs come from the 8086 CPU. They must remain stable during the memory cycle for proper operation of the $\overline{\text{CAS}}$ outputs.
$\overline{\text{CE}}$	Chip enable. This input is used to select the memory and enable the hidden refresh logic.
ALE	Address latch enable. This input is used to indicate the beginning of a memory cycle.
RFCK	Refresh clock. The period of this input determines the refresh interval. The duty cycle of this clock will determine the length of time that the circuit will attempt a hidden refresh.
AWAIT	When connected to V_{CC}, the DP84332 will insert an extra wait state in selected memory cycles.
$\overline{\text{RFRQ}}$	Refresh request. This input requests the DP84332 to perform a refresh. The state of the RFCK input will determine what type of refresh will be performed.

Output Signals

RASIN	This output provides a memory cycle start signal to the DP8408, and provides $\overline{\text{RAS}}$ timing during refresh.
$\overline{\text{CASH}}$, $\overline{\text{CASL}}$	These signals are the separate $\overline{\text{CAS}}$s needed for byte writing. Their presence is controlled by $\overline{\text{BHE}}$ and A0 respectively.
RDY	This output is used to insert a wait state into the 8086 memory cycles when selected and during a forced refresh cycle where the 8086 attempts to access the memory. The 8284A clock circuit should be configured so that ASYNC is enabled.
RFSH	This output controls the mode of the DP8408 dynamic RAM controller. When low, it switches the DP8408 into an all $\overline{\text{RAS}}$ refresh mode. This signal is also used to reset the refresh request logic.

FUNCTIONAL DESCRIPTION

A memory cycle starts when chip select ($\overline{\text{CS}}$) and the address latch enable (ALE) are true. RASIN is supplied from the DP84332 to the DP8408 dynamic RAM controller which then supplies a RAS signal to the selected dynamic RAM bank. After the necessary row address hold time, the DP8408 switches the address outputs to the column address. The DP84332 then supplies the required $\overline{\text{CAS}}$ signals ($\overline{\text{CASH}}$, $\overline{\text{CASL}}$) to the RAM. For byte operations, only one $\overline{\text{CAS}}$ will be activated. To differentiate between a read and a write, the $\text{DT}/\overline{\text{R}}$ signal from the CPU is inverted and supplied by the DP8408 to the memory array.

A refresh cycle is started by one of two conditions. One is when a refresh is requested (RFRQ is true), refresh clock (RFCK) is high, and a non-selected memory cycle is started (CE is not true, ALE is high). This is called hidden refresh because it is transparent to the CPU. In this case, the address supplied to the memories comes from the refresh counter in the DP8408, and no $\overline{\text{CAS}}$ signals are generated from the DP84332. The second form of refresh occurs when a refresh is requested, refresh clock is low, and there is no memory cycle in progress. This is called forced refresh, because the CPU will be forced to wait during the next memory cycle to allow for the refresh to be performed. In this case, a refresh is performed as before, but any attempt to access memory is delayed by wait states until after the refresh is finished. In either case, the refresh request is cleared by the refresh line ($\overline{\text{RFSH}}$), which also goes to the DP8408.

In a standard memory cycle, the access can be slowed down by one clock cycle to accommodate slower memories. This extra wait state will not appear during the hidden refresh cycle, so faster devices on the CPU bus will not be affected.

With higher speed systems, memory speed requirements will affect the performance of the system. Table 7.21.1 shows memory speed requirements at three different CPU clock speeds.

TABLE 7.21.1. Memory Speed Requirements

CPU Clock Frequency	t_{CAC}		t_{RAH}
	No Wait States	1 Wait State	
8 MHz	≤105 ns	≤223 ns	≤30 ns
5 MHz	≤170 ns	≤370 ns	≤30 ns

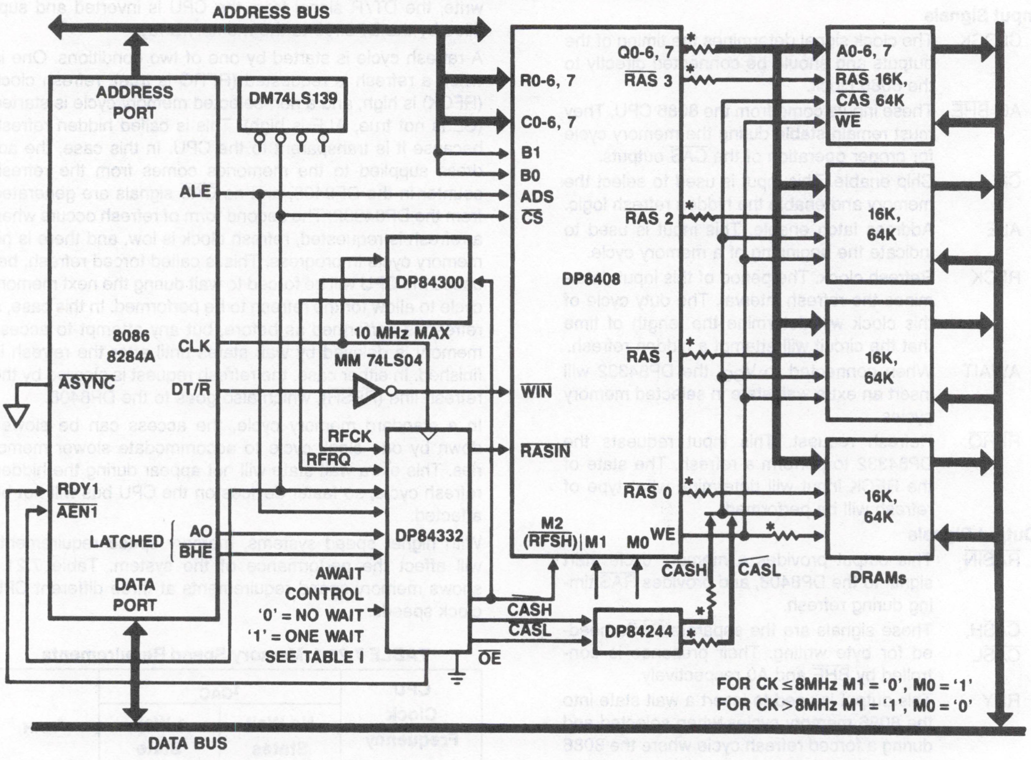
SYSTEM DESCRIPTION

For memory operation, the DP84332 can be directly connected between the control signals from the CPU chip set and the DP8408 dynamic RAM controller. Each CAS output of the DP84332 is capable of driving eight memory devices. If additional drive is required, a DP84244 buffer can be used to increase the fanout to the full capabilities of the DP8408 (eight memories per output of the DP84244).

The 84332 is a standard National Semiconductor PAL part (PAL 16R8). The user can modify the PAL equations to support this particular application. The 84332 logic equations, function table, and logic diagram can be seen at the end of this section.

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)

Interfacing The DP8408 to an 8086 System


ADDRESS BUS
ADDRESS PORT
DM74LS139
ALE
8086
8284A
CLK
ASYNC
DT/R̄
RDY1
AEN1
LATCHED
AO
BHE
DATA PORT
DP84300
10 MHz MAX
MM 74LS04
RFCK
RFRQ
DP84332
AWAIT
CONTROL
'0' = NO WAIT
'1' = ONE WAIT
SEE TABLE I
CASH
CASL
OE
R0-6, 7
C0-6, 7
B1
B0
ADS
CS
WIN
RASIN
DP8408
M2
(RFSH)
M1
M0
Q0-6, 7
RAS 3
RAS 2
RAS 1
RAS 0
WE
CASH
CASL
DP84244
A0-6, 7
RAS 16K,
CAS 64K
WE
16K,
64K
DRAMs
FOR CK ≤ 8MHz M1 = '1', M0 = '1'
FOR CK > 8MHz M1 = '1', M0 = '0'
DATA BUS
*

*These outputs may need resistors.

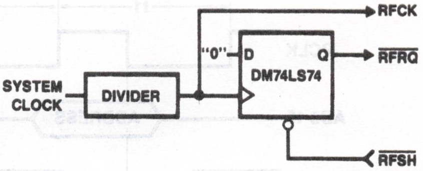
TL/L/9991-B5

FIGURE 7.21.3. System Block Diagram

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)

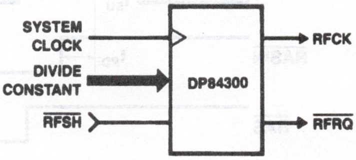
REFRESH REQUEST LOGIC

To generate the refresh request for the DP84332, external circuitry is required. *Figure 7.21.4* shows how this can be implemented, using standard SSI and MSI logic. A DM74LS393 counter is used to time the period between refresh cycles, while the DM74LS74 flip-flop is used to record the need of a new refresh. A better solution is to use the 24-pin DP84300 programmable refresh timer, as shown in *Figure 7.21.5*. This part allows a maximum amount of time for a hidden refresh to occur before lowering the refresh clock output, and implements the refresh request logic.


RFCK
"0"
D
Q
RFRQ
DM74LS74
SYSTEM CLOCK
DIVIDER
RFSH

TL/L/9991-B6

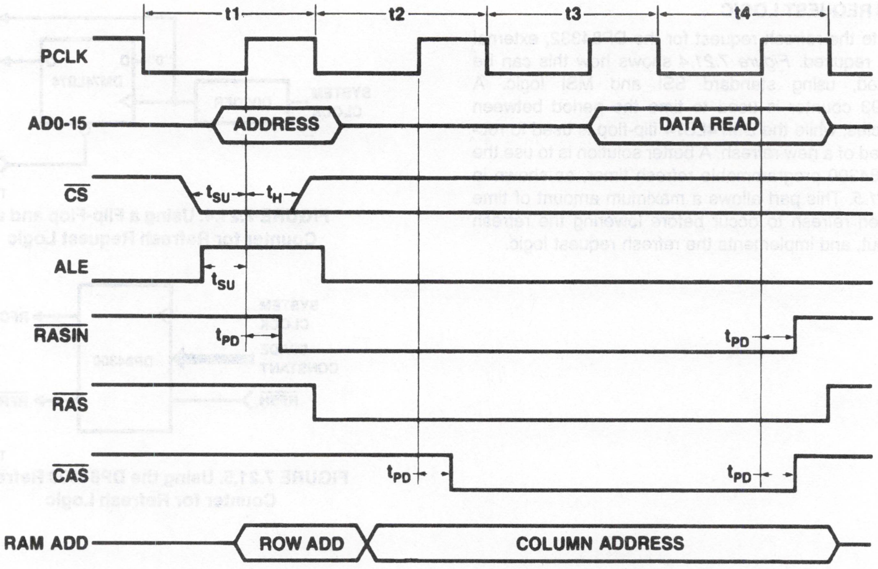
FIGURE 7.21.4. Using a Flip-Flop and a Counter for Refresh Request Logic


SYSTEM CLOCK
DIVIDE CONSTANT
RFSH
DP84300
RFCK
RFRQ

TL/L/9991-B7

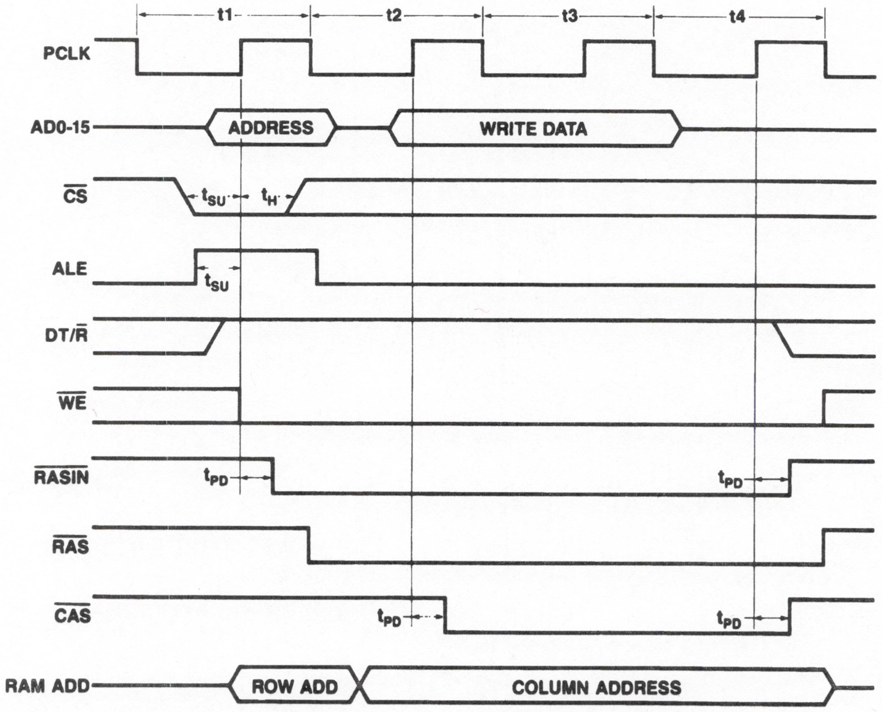
FIGURE 7.21.5. Using the DP84300 Refresh Counter for Refresh Logic

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)


t1
t2
t3
t4
PCLK
AD0-15
ADDRESS
DATA READ
CS
t_{SU}
t_H
ALE
t_{SU}
RASIN
t_{PD}
t_{PD}
RAS
CAS
t_{PD}
t_{PD}
RAM ADD
ROW ADD
COLUMN ADDRESS

TL/L/9991–B8

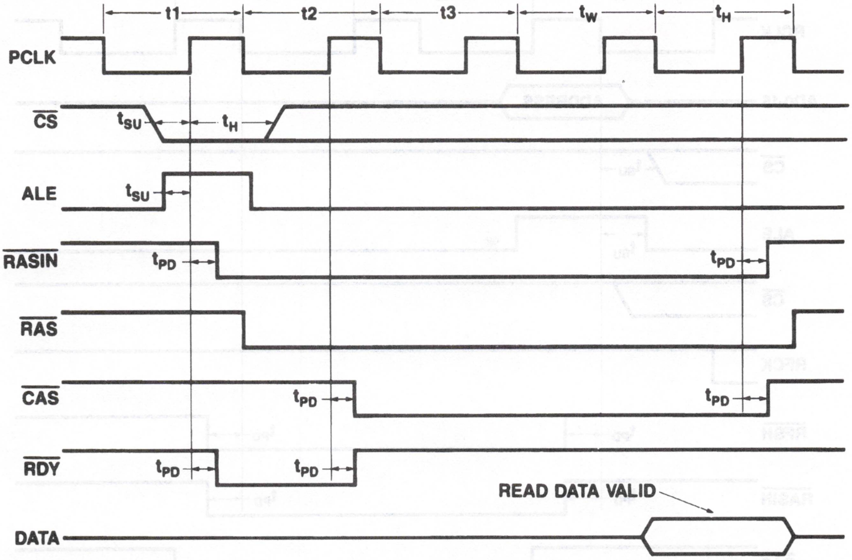
FIGURE 7.21.6. Timing Diagram; Read Timing


t1
t2
t3
t4
PCLK
AD0-15
ADDRESS
WRITE DATA
CS
t_{SU}
t_H
ALE
t_{SU}
DT/R
WE
RASIN
t_{PD}
t_{PD}
RAS
CAS
t_{PD}
t_{PD}
RAM ADD
ROW ADD
COLUMN ADDRESS

TL/L/9991–B9

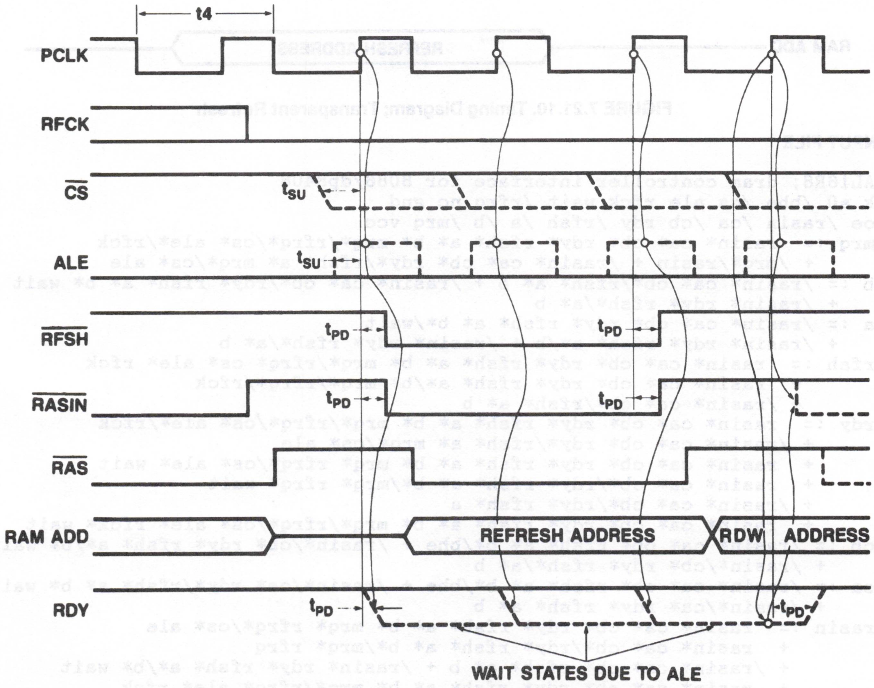
FIGURE 7.21.7. Timing Diagram; Write Timing

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)


t1
t2
t3
tW
tH
PCLK
CS
tSU
tH
ALE
tSU
RASIN
tPD
tPD
RAS
CAS
tPD
tPD
RDY
tPD
tPD
READ DATA VALID
DATA

TL/L/9991-C0

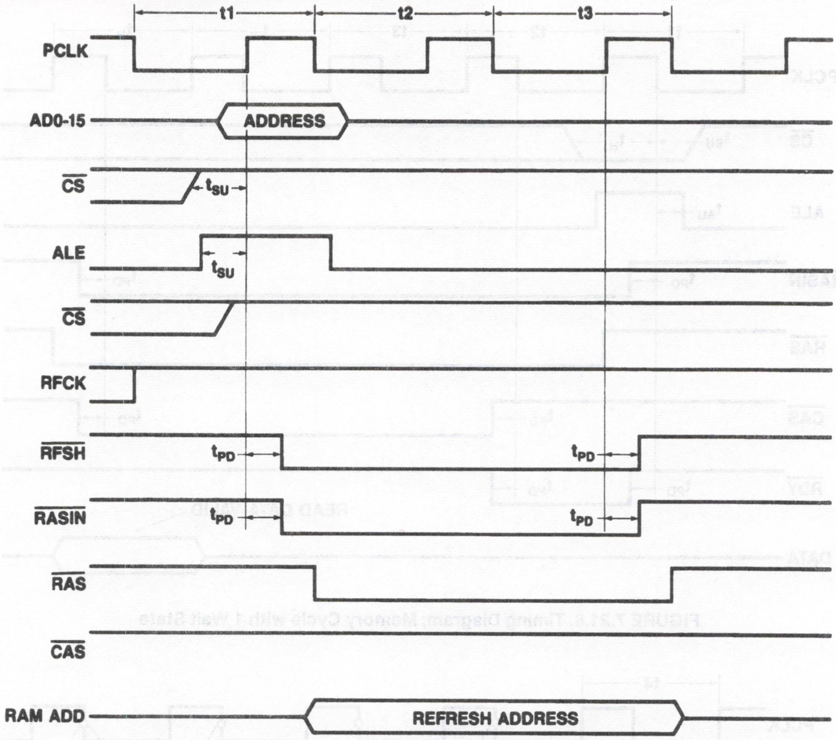
FIGURE 7.21.8. Timing Diagram; Memory Cycle with 1 Wait State


t4
PCLK
RFCK
CS
tSU
ALE
tSU
RFSH
tPD
tPD
RASIN
tPD
tPD
RAS
RAM ADD
REFRESH ADDRESS
RDW
ADDRESS
RDY
tPD
tPD
WAIT STATES DUE TO ALE

TL/L/9991-C1

FIGURE 7.21.9. Timing Diagram; Forced Refresh

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)



TL/L/9991-C2

FIGURE 7.21.10. Timing Diagram; Transparent Refresh

PLAN™ INPUT FILE

```
PAL16R8; dram controller interface for 8086/dp8408
ck a0 /bhe /cs ale rfck wait /rfrq nc gnd
/oe /rasin /ca /cb rdy /rfsh /a /b /mrq vcc
/mrq :=  rasin* ca* cb* rdy* rfsh* a* b* mrq*/rfrq*/cs* ale*/rfck
      + /mrq*/rasin + /rasin* ca* cb* rdy*/rfsh* a* mrq*/cs* ale
/b := /rasin* ca* cb*/rfsh* a* b + /rasin* ca* cb*/rdy* rfsh* a* b* wait
    + /rasin* rdy* rfsh*/a* b
/a := /rasin* ca* cb* rdy* rfsh* a* b*/wait
    + /rasin* rdy* rfsh* a*/b + /rasin* rdy* rfsh*/a* b
/rfsh :=  rasin* ca* cb* rdy* rfsh* a* b* mrq*/rfrq* cs* ale* rfck
       +  rasin* ca* cb* rdy* rfsh* a*/b* mrq*/rfrq*/rfck
       + /rasin* ca* cb*/rfsh* a* b
/rdy :=  rasin* ca* cb* rdy* rfsh* a* b* mrq*/rfrq*/cs* ale*/rfck
      + /rasin* ca* cb* rdy*/rfsh* a* mrq*/cs* ale
      +  rasin* ca* cb* rdy* rfsh* a* b* mrq* rfrq*/cs* ale* wait
      +  rasin* ca* cb*/rdy* rfsh* a* b*/mrq* rfrq* wait
      + /rasin* ca* cb*/rdy* rfsh* a
      +  rasin* ca* cb* rdy* rfsh* a* b* mrq*/rfrq*/cs* ale* rfck* wait
/cb := /rasin* ca* cb* rfsh* a* b*/bhe + /rasin*/cb* rdy* rfsh* a*/b* wait
     + /rasin*/cb* rdy* rfsh*/a* b
/ca := /rasin* ca* cb* rfsh* a* b*/bhe + /rasin*/ca* rdy*/rfsh* a* b* wait
     + /rasin*/ca* rdy* rfsh* a* b
/rasin :=  rasin* ca* cb* rdy* rfsh* a* b* mrq* rfrq*/cs* ale
        +  rasin* ca* cb*/rdy* rfsh* a* b*/mrq* rfrq
        + /rasin* ca* cb* rfsh* a* b + /rasin* rdy* rfsh* a*/b* wait
        +  rasin* ca* cb* rdy* rfsh* a* b* mrq*/rfrq* ale* rfck
        +  rasin* ca* cb* rdy* rfsh* a* b* mrq*/rfrq*/rfck
        + /rasin* ca* cb*/rfsh* a* b + /rasin* rdy* rfsh*/a* b
```

TL/L/9991-H1

7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)

PLAN™ JEDEC FILE

```
PLAN v3.12   08-31-1988    08:22
Source filename: DP84332.BEQ    Device: PAL16R8
dram controller interface for 8086/dp8408 *
QP20* QF2048* F0*
L0000
11011101100101011001110110011101
11101111111111111111111111111110
11011111100101101101110111011110*
L0256
11111101110111101111110111011110
11111101110111011110010111011110
11111101111011011101111111111110*
L0512
11111101110111011101100111011110
11111110110111011101111111111110
11111101111011011101111111111110*
L0768
11011101010101010101110110011101
11011110110111011001110110011101
11111101110111101111110111011110*
L1024
11011101100101011001110110011101
11011111100101101101110111011110
11011101100101011101010101011101
11101101110111011110010101011101
11111111110111011110110111011110
11011101100101010101010110011101*
L1280
11111001110111011111110111011110
11111110110111011101011011111110
11111101111011011101111011111110*
L1536
11111001110111011111110111011110
11111101110111101101011111101110
11111101110111011101111111101110*
L1792
11011101100101011101110101011101
11101101110111011110110101011101
11111101110111011111110111011110
11111110110111011101011111111110
11011101110101010101110110011101
11011101110111011001110110011101
11111101110111101111110111011110
11111101111011011101111111111110*
C5748* FA0E
```

TL/L/9991-H2

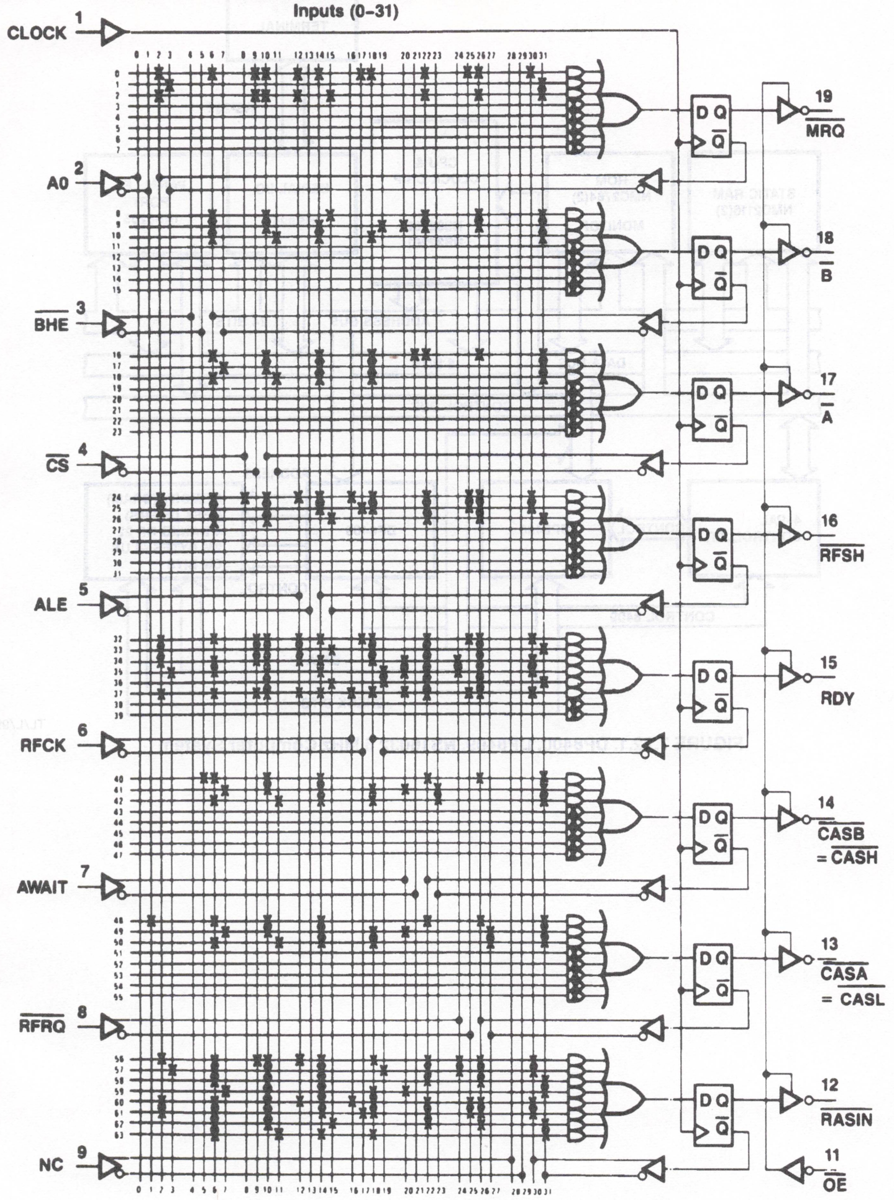
7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)

TABLE 7.21.2. Function Table

CK	A0	$\overline{\text{BHE}}$	$\overline{\text{CS}}$	ALE	RFCK	WAIT	$\overline{\text{RFRQ}}$	$\overline{\text{OE}}$	$\overline{\text{RASIN}}$	$\overline{\text{CA}}$	$\overline{\text{CB}}$	RDY	$\overline{\text{RFSH}}$	$\overline{\text{A}}$	$\overline{\text{B}}$	$\overline{\text{MRQ}}$
C	L	L	H	L	H	L	H	L	X	X	X	X	X	X	X	X
C	L	L	H	L	H	L	H	L	H	H	H	H	H	H	H	H
C	X	X	L	H	X	L	H	L	L	H	H	H	H	H	H	H
C	L	H	L	L	X	L	H	L	L	L	H	H	H	L	H	H
C	L	H	L	L	X	L	H	L	L	H	H	H	H	L	L	H
C	X	X	H	L	H	L	H	L	H	H	H	H	H	H	H	H
C	X	X	L	H	X	H	H	L	L	H	H	L	H	H	H	H
C	H	H	L	L	X	H	H	L	L	H	H	H	H	H	L	H
C	H	H	L	L	X	H	H	L	L	H	H	H	H	L	H	H
C	H	H	L	L	X	H	H	L	L	H	H	H	H	L	L	H
C	X	X	H	L	H	H	H	L	H	H	H	H	H	H	H	H
C	X	X	H	H	H	X	L	L	L	H	H	H	L	H	H	H
C	X	X	H	L	X	X	X	L	L	H	H	H	L	H	L	H
C	X	X	X	L	X	X	H	L	H	H	H	H	H	H	H	H
C	X	X	X	L	L	X	L	L	L	H	H	H	L	H	H	H
C	X	X	X	L	X	X	X	L	L	H	H	H	L	H	L	H
C	X	X	X	L	X	X	H	L	H	H	H	H	H	H	H	H
C	X	X	L	H	L	X	L	L	L	H	H	L	L	H	H	L
C	H	L	L	L	X	X	X	L	L	H	H	L	L	H	L	L
C	H	L	L	L	X	X	H	L	H	H	H	L	H	H	H	L
C	H	L	L	L	X	L	H	L	L	H	H	H	H	H	H	H
C	H	L	L	L	X	L	H	L	L	H	L	H	H	L	H	H
C	H	L	L	L	X	L	H	L	L	H	L	H	H	L	L	H
C	X	X	L	L	X	L	H	L	H	H	H	H	H	H	H	H
C	X	X	X	L	L	X	L	L	L	H	H	H	L	H	H	H
C	X	X	L	H	X	H	X	L	L	H	H	L	L	H	L	L
C	L	L	L	L	X	H	X	L	H	H	H	L	H	H	H	L
C	L	L	L	L	X	H	H	L	L	H	H	L	H	H	H	H
C	L	L	L	L	X	H	H	L	L	L	L	H	H	H	L	H
C	L	L	L	L	X	H	H	L	L	L	L	H	H	L	H	H
C	L	L	L	L	X	H	H	L	L	L	L	H	H	L	L	H
C	X	X	L	L	X	H	H	L	H	H	H	H	H	H	H	H
C	X	X	L	H	H	L	L	L	L	H	H	H	H	H	H	H
C	H	L	L	L	X	L	X	L	L	H	L	H	H	L	H	H
C	H	L	L	L	X	L	X	L	L	H	L	H	H	L	L	H
C	X	X	L	L	X	L	X	L	H	H	H	H	H	H	H	H
C	X	X	L	H	H	H	L	L	L	H	H	L	H	H	H	H
C	X	X	L	L	X	X	X	H	Z	Z	Z	Z	Z	Z	Z	Z

TL/L/9991–C5

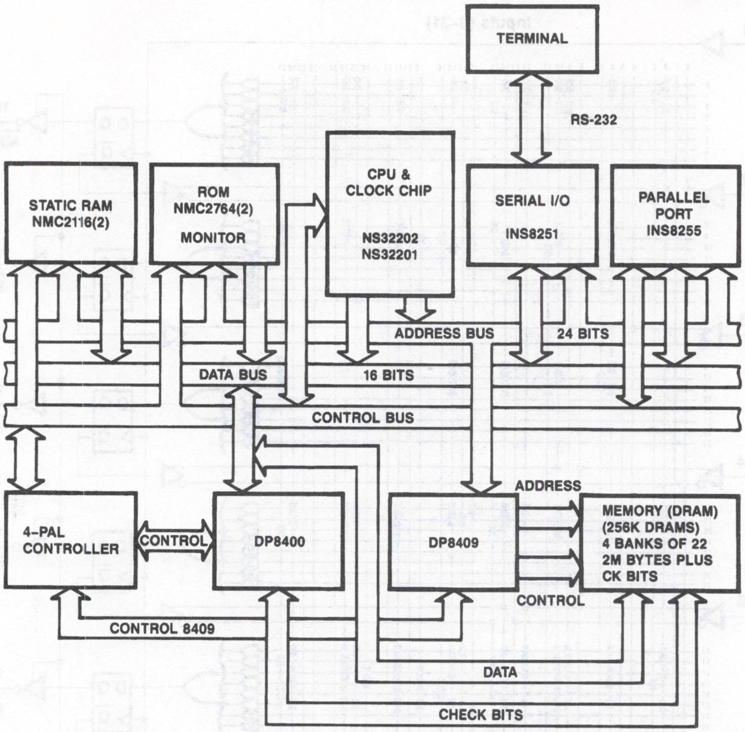
7.21 DP84332 Dynamic RAM Controller Interface Circuit for the 8086 and 8088 CPUs (Continued)


Inputs (0-31)
1 CLOCK
2 A0
3 BHE
4 CS
5 ALE
6 RFCK
7 AWAIT
8 RFRQ
9 NC
19 MRQ
18 B
17 A
16 RFSH
15 RDY
14 CASB = CASH
13 CASA = CASL
12 RASIN
11 OE
D Q
Q

TL/L/9991–C6

FIGURE 7.21.11. PAL16R8 Logic Diagram Showing DP84332 Pattern

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032*


TERMINAL
RS-232
STATIC RAM
NMC2116(2)
ROM
NMC2764(2)
MONITOR
CPU &
CLOCK CHIP
NS32202
NS32201
SERIAL I/O
INS8251
PARALLEL
PORT
INS8255
ADDRESS BUS
24 BITS
DATA BUS
16 BITS
CONTROL BUS
ADDRESS
4-PAL
CONTROLLER
CONTROL
DP8400
DP8409
MEMORY (DRAM)
(256K DRAMS)
4 BANKS OF 22
2M BYTES PLUS
CK BITS
CONTROL
CONTROL 8409
DATA
CHECK BITS

TL/L/9991-C7

FIGURE 7.22.1. DP8400, DP8409, NS16032 6 MHz Computer System

---

*Applications contained in this section are for illustration purposes only and National makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

Four PAL devices were used in this application in order to interface between the NS32032, DP8409 and the DP8400. These PAL devices have the following features:

1. The PAL devices control the following types of cycles:
   a) READ cycles with no errors detected, ALWAYS CORRECT MODE
   b) READ cycles with single error detected, the correct data will be written back to memory
   c) WRITE cycles
   d) BYTE WRITE cycles
   e) DRAM REFRESH cycles

   The PAL devices take care of everything, no extra control logic is needed.
2. The outputs of the PAL device control the DP8409, the DP8400 and insert WAIT states at the appropriate times into the NS32032 cycles.
3. The PAL device contains outputs to interrupt the NS32032, or cause a cycle abort if an error greater than a single error is detected (DOUBLERR), or if there is a bus parity error in data transfer from the CPU to memory (PARITYERROR).
4. This PAL device design should work up to 8 MHz with the NS32032. If it is desired to go faster, another WAIT state will have to be inserted into all cycles, and the PAL device equations will have to be adjusted accordingly. Another possibility would be to use the new oxide isolated DP8400 and the new DRAM controller DP8419 (pin compatible with DP8409 in modes 0, 1, 4, 5). These parts would allow considerably more time margin.
5. As can be seen by looking at the PAL device logic diagrams some external logic is needed and some external logic may be added. For example, a system reset input could be added to allow the internal flip-flops to be set to a known state—in this case a refresh state (In PAL device number 1, for example, I used external logic to "NOR" the RFI/O input with a system RESET input). An output enable input was also included to allow all the PAL device outputs to be TRI-STATE outputs.
6. This PAL device interface performs HIDDEN REFRESHES (CPU not accessing the Dynamic RAM controlled by the DP8409, indicated by /CS being high) assuming a four-T state processor access cycle.
7. Logic diagrams, the PAL device equations, and the timing diagrams follow this introduction section. Basically everything is self-explanatory.
8. I feel that if one is using this interface above 4–6 MHz, he should use the fast PAL devices (example "PAL16R8A" instead of "PAL16R8"). The fast PAL devices have an input to output maximum time of 25 ns and 15 ns if it is a registered output.

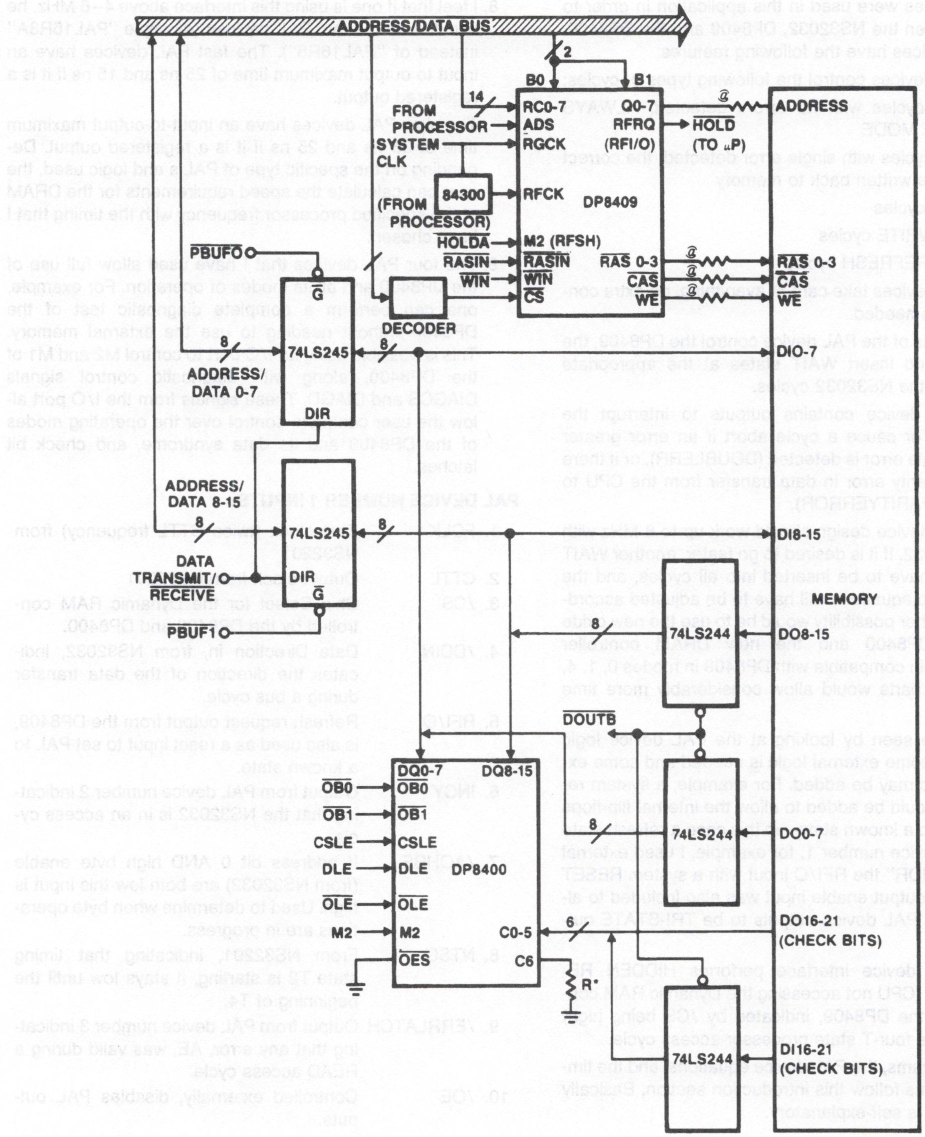
   The slow PAL devices have an input-to-output maximum time of 35 ns and 25 ns if it is a registered output. Depending on the specific type of PAL's and logic used, the user can calculate the speed requirements for the DRAM at the specified processor frequency with the timing that I have chosen.
9. The four PAL devices that I have used allow full use of the DP8400 and all its modes of operation. For example, one can perform a complete diagnostic test of the DP8400 without needing to use the external memory. This is possible using an I/O port to control M2 and M1 of the DP8400, along with diagnostic control signals DIAGCS and DIAGD. These signals from the I/O port allow the user complete control over the operating modes of the DP8400 and its data syndrome, and check bit latches.

PAL DEVICE NUMBER 1 INPUTS

1.	FCLK	Fast clock (twice CTTL frequency) from NS32201
2.	CTTL	Output clock from NS32201
3.	/CS	Chip Select for the Dynamic RAM controlled by the DP8409 and DP8400.
4.	/DDIN	Data Direction in, from NS32032, indicates the direction of the data transfer during a bus cycle.
5.	RFI/O	Refresh request output from the DP8409, is also used as a reset input to set PAL to a known state.
6.	INCY	Output from PAL device number 2 indicating that the NS32032 is in an access cycle.
7.	/AOHBE	If address bit 0 AND high byte enable (from NS32032) are both low this input is high. Used to determine when byte operations are in progress.
8.	NTSO	From NS32201, indicating that timing state T2 is starting, it stays low until the beginning of T4.
9.	/ERRLATCH	Output from PAL device number 3 indicating that any error, AE, was valid during a READ access cycle.
10.	/OE	Controlled externally, disables PAL outputs.

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)


ADDRESS/DATA BUS
2
B0
B1
14
FROM PROCESSOR
SYSTEM CLK
RC0-7
ADS
RGCK
Q0-7
RFRQ
(RFI/O)
HOLD
(TO μP)
@
ADDRESS
84300
(FROM PROCESSOR)
RFCK
DP8409
HOLDA
RASIN
WIN
M2 (RFSH)
RASIN
WIN
CS
RAS 0-3
CAS
WE
RAS 0-3
CAS
WE
PBUF0
G
DECODER
8
74LS245
8
ADDRESS/ DATA 0-7
DIR
DI0-7
ADDRESS/ DATA 8-15
8
74LS245
8
DI8-15
DATA TRANSMIT/ RECEIVE
DIR
G
MEMORY
PBUF1
8
74LS244
DO8-15
DOUTB
DQ0-7
DQ8-15
OB0
OB1
CSLE
DLE
OLE
M2
OB0
OB1
CSLE
DLE
DP8400
OLE
M2
OES
C0-5
C6
8
74LS244
DO0-7
6
DO16-21 (CHECK BITS)
R*
74LS244
DI16-21 (CHECK BITS)

TL/L/9991–C8

@Resistor required depends on DRAM load.
*R = 2.7 kΩ

FIGURE 7.22.2. DP8400/8409 System Interface Block Diagram

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PAL DEVICE NUMBER 1 OUTPUTS

1. /RASIN	Input to DP84909.
2. /RFSH	Input to DP8409, causes the DP8409 to enter mode 1 to do a refresh.
3. /1DLY	Delay used by the PAL devices to determine the state of the processor system.
4. /2DLY	Delay used by the PAL devices to determine the state of the processor system.
5. /3DLY	Delay used by the PAL devices to determine the state of the processor system.
6. /4DLY	Delay used by the PAL devices to determine the state of the processor system.
7. /ODCLEN	/OLE, DLE, CSLE enable latch signal.
8. /CYCLED	Indicates that a processor access cycle is complete.

PAL DEVICE NUMBER 2 INPUTS

1. /RFSH	Output from PAL device number 1 that indicates whether the DRAMs are being refreshed.
2. /RASIN	Output from PAL device number 1.
3. A0	Output from NS32032, address bit 0.
4. /HBE	Output from NS32032, high byte enable.
5. /DDIN	Data Direction in, from NS32032.
6. /ADS	Address strobe from NS32032.
7. NTSO	Output from NS32201.
8. /2DLY	Output from PAL device number 1.
9. /4DLY	Output from PAL device number 1.
10. /ERRLATCH	Output from PAL device number 3 indicating that an error has occurred during a READ cycle.
11. CSOE	Chip select Output Enable, disable the outputs of the PAL device when low, and also used for other control purposes.

PAL DEVICE NUMBER 2 OUTPUTS

1. /0B0	Control DP8400 output buffer for byte "0".
2. 0B1	Controls DP8400 output buffer for byte "1".
3. /PBUF0	Controls the processor buffer transceiver for byte "0".
4. /PBUF1	Controls the processor buffer transceiver for byte "1"
5. /DOUTB	Controls memory buffers that interface between the DRAM and the DP8400/memory data bus.
6. /INCY	Output indicating that the NS32032 is in an access cycle.
7. /CWAIT	Output to NS32201 that causes WAIT states to be inserted into the NS32032 bus cycles.

PAL DEVICE NUMBER 3 INPUTS

1. /DDIN	Output from NS32032.
2. /RFSH	Output from PAL device number 1 indicating a forced refresh of the memory.
3. /AOHBE	Output of A0 and /HBE logically NORed together. Therefore, if either input is high this signal will be low. This signal is useful to determine whether words or bytes are being written.
4. /ERRLATCH	Output from PAL device number 4 indicating that an error has occurred during a CS READ cycle, it may be a single or multiple bit error.
5. /1DLY	Input from PAL device number 1.
6. /2DLY	Input from PAL device number 1.
7. /3DLY	Input from PAL device number 1.
8. /4DLY	Input from PAL device number 1.
9. /RESET	Input from external logic that resets the double bit error latch /DOUBLERR or the parity error latch PARITYERR.
10. AE	Output from DP8400 indicating an error.
11. EO	Output from DP8400 indicating the type of error.
12. E1	Output from DP8400 indicating the type of error.
13. /PARITYERROR	This is an output of this PAL device also. This input indicates that a PARITY error has occurred during a WRITE cycle.
14. CSOE	Chip Select Output enable, disables the registered outputs of the PAL device when low.

PAL DEVICE NUMBER 3 OUTPUTS

1. /WIN	Input to the DP8409.
2. /MODECC	Input to the DP8400, changes between READ and WRITE modes.
3. /PARITYERR	Can be used to interrupt the system when a parity error has been detected during a WRITE cycle.

PAL DEVICE NUMBER 4 INPUTS

1. FCLK	Fast clock from NS32201.
2. ODCLEN	/OLE, DLE, CSLE latch enable input.
3. DIAGCS	Enable input from I/O port for diagnostics to enable CSLE, check bit syndrome latch enable.
4. DIAGD	Enable input from I/O port for diagnostics to enable DLE, data latch enable.
5. /RESET	Reset input from I/O port to reset PAL error latches.

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PAL DEVICE NUMBER 4 INPUTS (Continued)

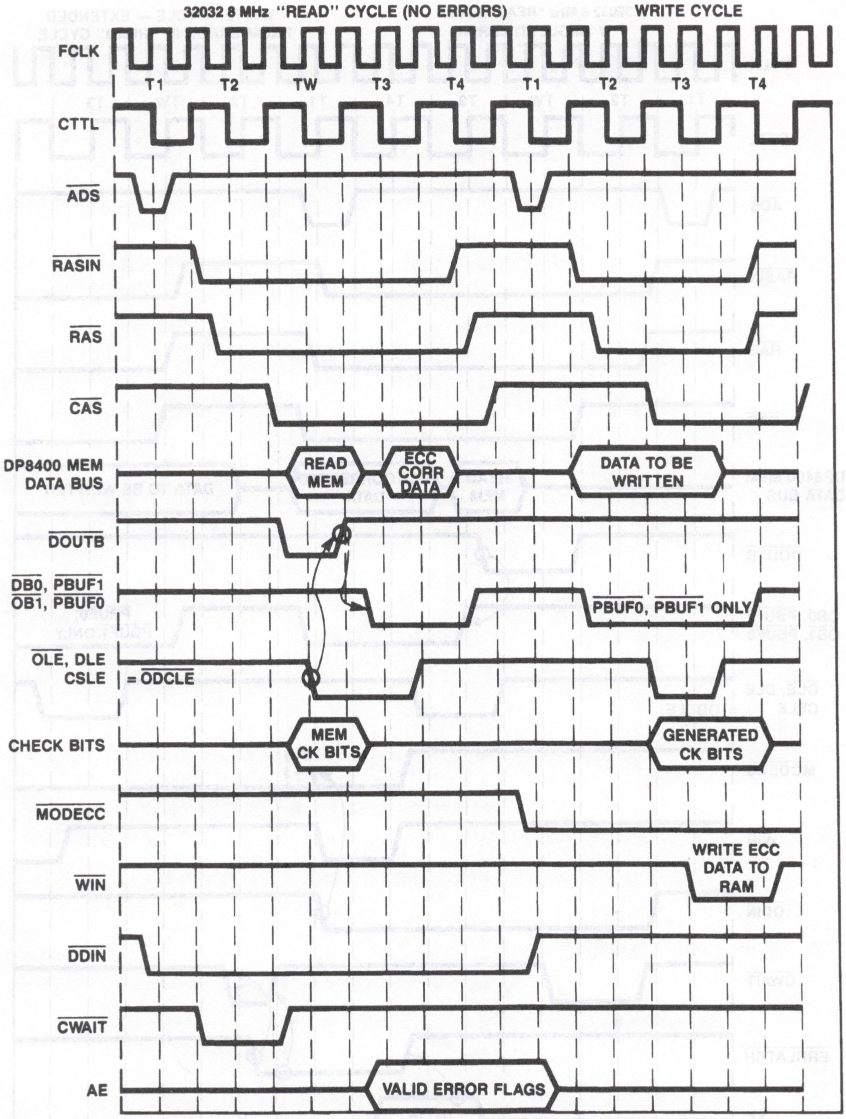
6. /CYCLED	Output from PAL device number 1 indicating that a processor access cycle is complete.
7. AE	Output from DP8400 indicating an error.
8. /E01	When this input is low it indicates that either error flag E0 or E1 was high.
9. /3DLY	This is an input from PAL device number 1.
10. /0E	Output from I/O port that enables the PAL outputs.
11. /DDIN	NS32032 input that indicates the direction of the bus transfer during a bus cycle.
12. /RFSH	Output from PAL device number 1 indicating a DRAM refresh cycle.

PAL DEVICE NUMBER 4 OUTPUTS

1. DLE	Output that controls the DP8400 Data latch.
2. CSLE	Output that controls the DP8400 Check bit Syndrome latch.
3. /OLE	Output that controls the DP8400 Output latch.
4. /DOUBLERR	Can be used to interrupt the system when a double bit error has been detected during a READ cycle.
5. /ERRLATCH	Used in the PAL device controller to indicate that an error has occurred during a /CS READ cycle, as indicated by AE being valid.

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

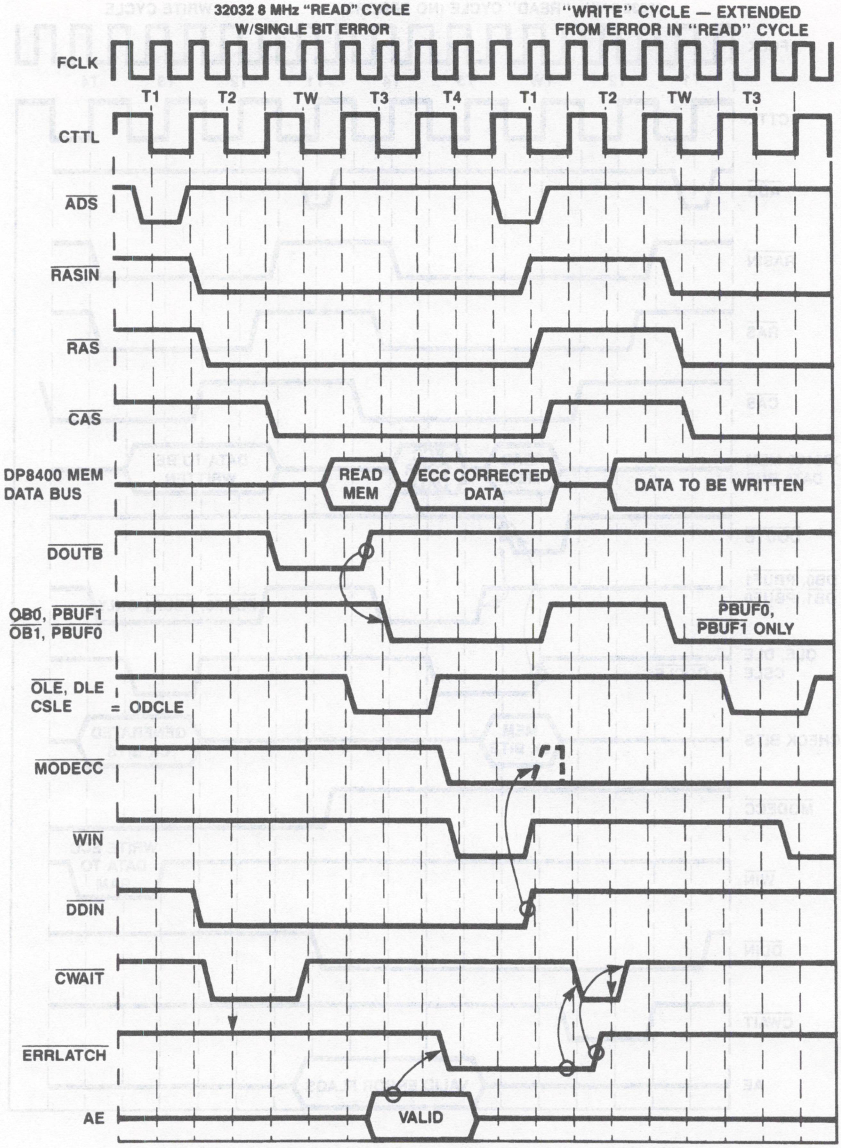

32032 8 MHz "READ" CYCLE (NO ERRORS)
WRITE CYCLE
FCLK
T1
T2
TW
T3
T4
T1
T2
T3
T4
CTTL
ADS
RASIN
RAS
CAS
DP8400 MEM DATA BUS
READ MEM
ECC CORR DATA
DATA TO BE WRITTEN
DOUTB
DB0, PBUF1
OB1, PBUF0
PBUF0, PBUF1 ONLY
OLE, DLE
CSLE
= ODCLE
CHECK BITS
MEM CK BITS
GENERATED CK BITS
MODECC
WRITE ECC DATA TO RAM
WIN
DDIN
CWAIT
AE
VALID ERROR FLAGS

TL/L/9991-C9

FIGURE 7.22.3. Timing Diagram; Read Cycle and Write Cycle

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

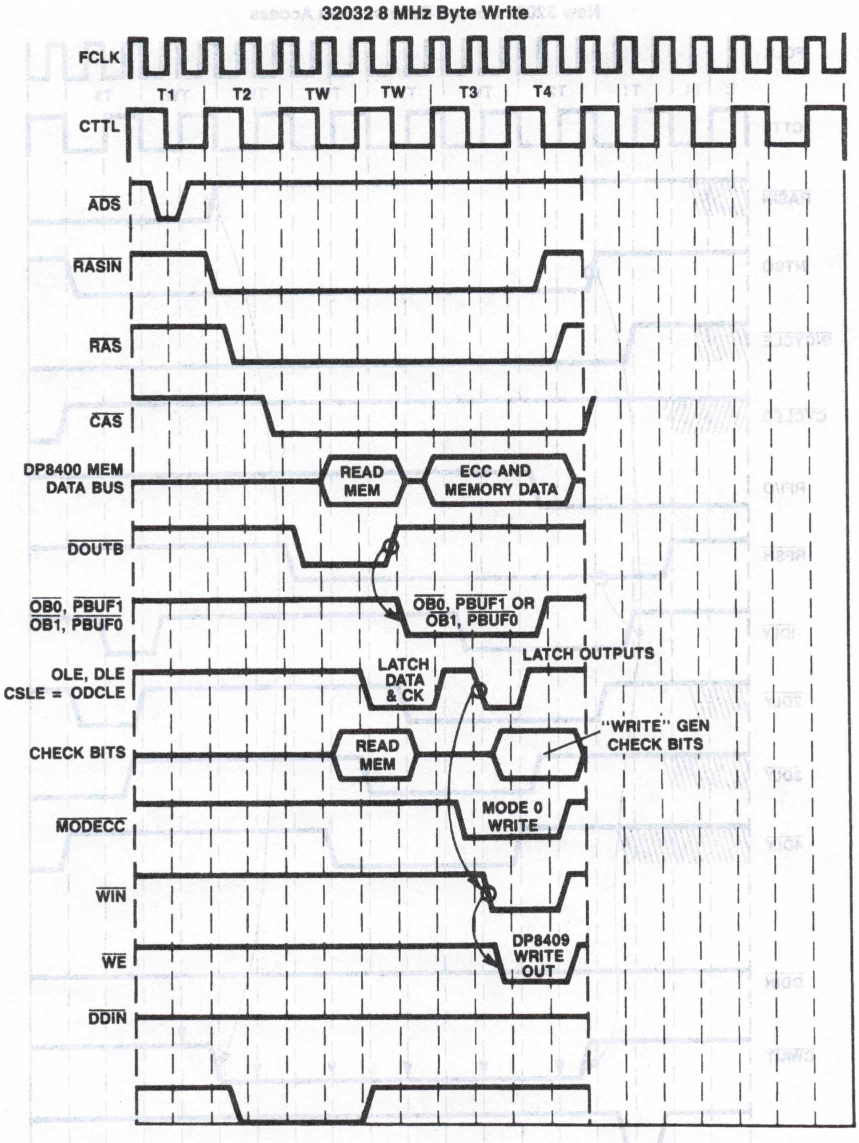

32032 8 MHz "READ" CYCLE
W/SINGLE BIT ERROR
"WRITE" CYCLE — EXTENDED
FROM ERROR IN "READ" CYCLE
FCLK
T1
T2
TW
T3
T4
T1
T2
TW
T3
CTTL
ADS
RASIN
RAS
CAS
DP8400 MEM
DATA BUS
READ
MEM
ECC CORRECTED
DATA
DATA TO BE WRITTEN
DOUTB
OB0, PBUF1
OB1, PBUF0
PBUF0,
PBUF1 ONLY
OLE, DLE
CSLE
= ODCLE
MODECC
WIN
DDIN
CWAIT
ERRLATCH
AE
VALID

TL/L/9991-D0

FIGURE 7.22.4. Timing Diagram; Read Cycle with Simple Bit Error

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)


32032 8 MHz Byte Write
FCLK
T1
T2
TW
TW
T3
T4
CTTL
ADS
RASIN
RAS
CAS
DP8400 MEM DATA BUS
READ MEM
ECC AND MEMORY DATA
DOUTB
OB0, PBUF1 OB1, PBUF0
OB0, PBUF1 OR OB1, PBUF0
OLE, DLE CSLE = ODCLE
LATCH DATA & CK
LATCH OUTPUTS
CHECK BITS
READ MEM
"WRITE" GEN CHECK BITS
MODECC
MODE 0 WRITE
WIN
WE
DP8409 WRITE OUT
DDIN

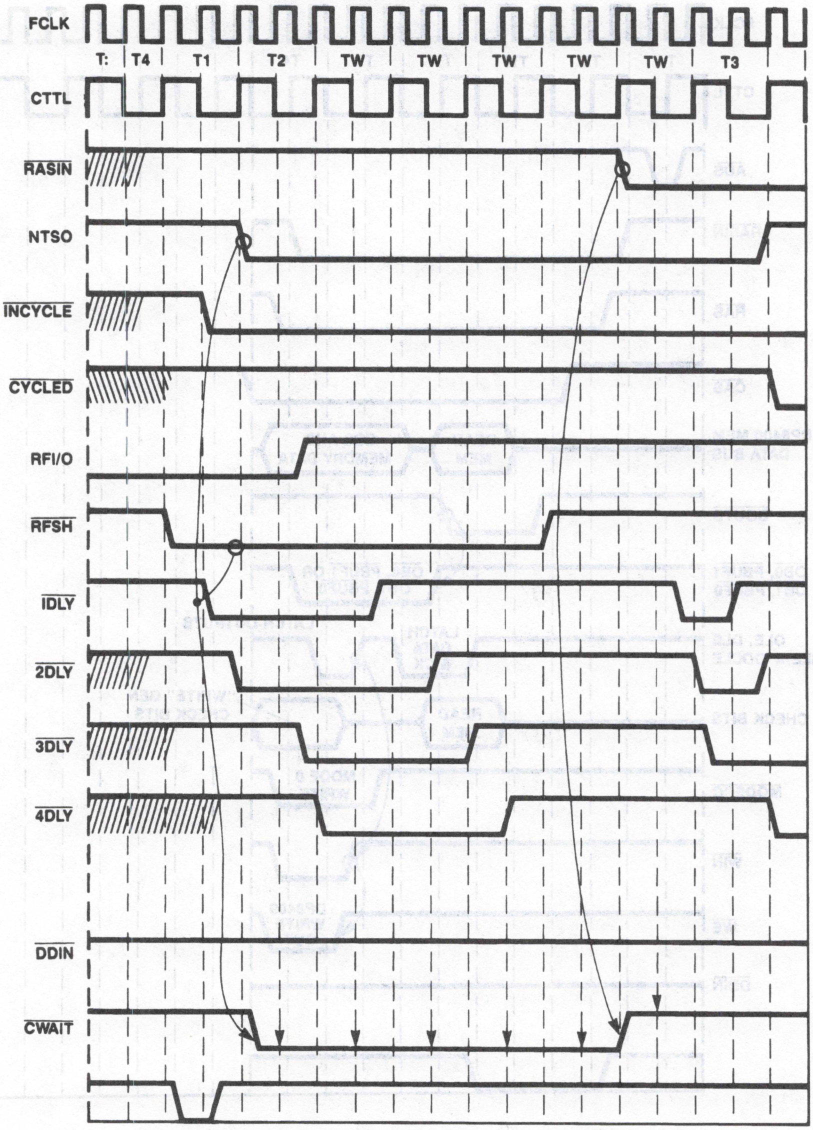
TL/L/9991-D1

FIGURE 7.22.5. Timing Diagram; Byte Write

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

New 32032 Forced Refresh then Access

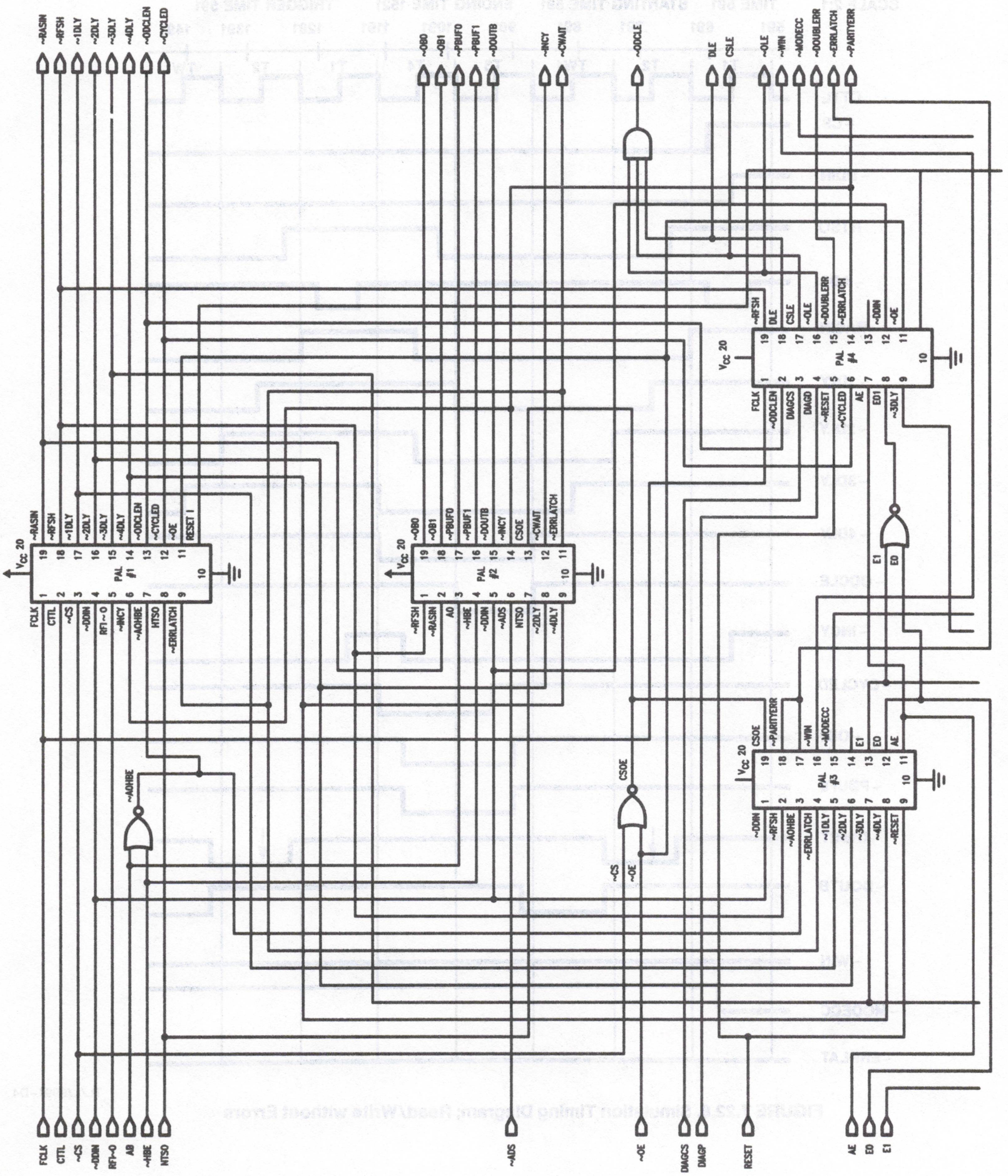

FCLK
T:
T4
T1
T2
TW
TW
TW
TW
TW
T3
CTTL
RASIN
NTSO
INCYCLE
CYCLED
RFI/O
RFSH
IDLY
2DLY
3DLY
4DLY
DDIN
CWAIT

TL/L/9991-D2

FIGURE 7.22.6. Timing Diagram; Forced Refresh then Access

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

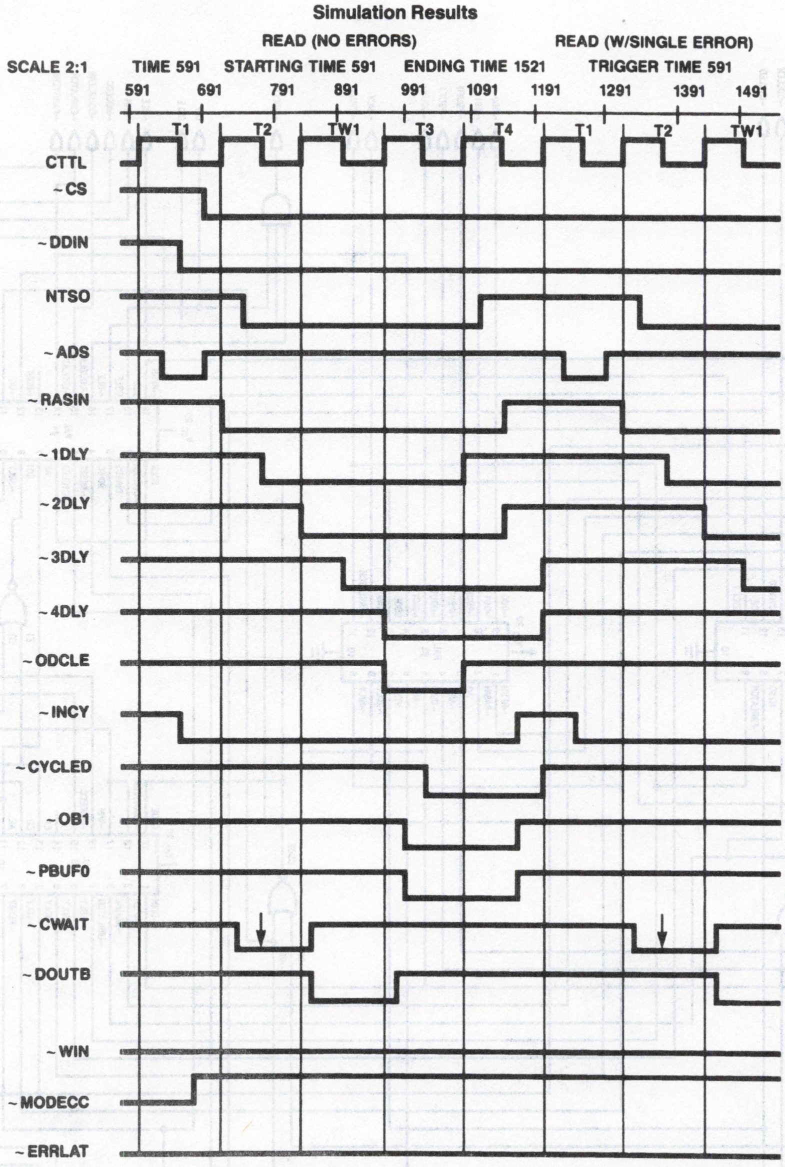

~RASIN
~RFSH
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLEN
~CYCLED
~OB0
~OB1
~PBUF0
~PBUF1
~DOUTB
~INCY
~CWAIT
~ODCLE
DLE
CSLE
~OLE
~WIN
~MODECC
~DOUBLERR
~ERRLATCH
~PARITYERR
PAL #1
PAL #2
PAL #3
PAL #4
VCC 20
FCLK
CTTL
~CS
~DDIN
RFI~O
A0
~HBE
NTSO
~AOHBE
~ADS
~OE
CSOE
DIAGCS
DIAGP
RESET
AE
EO
E1
ED1
DIAGD

TL/L/9991-D3

FIGURE 7.22.7. Simulation Circuit

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

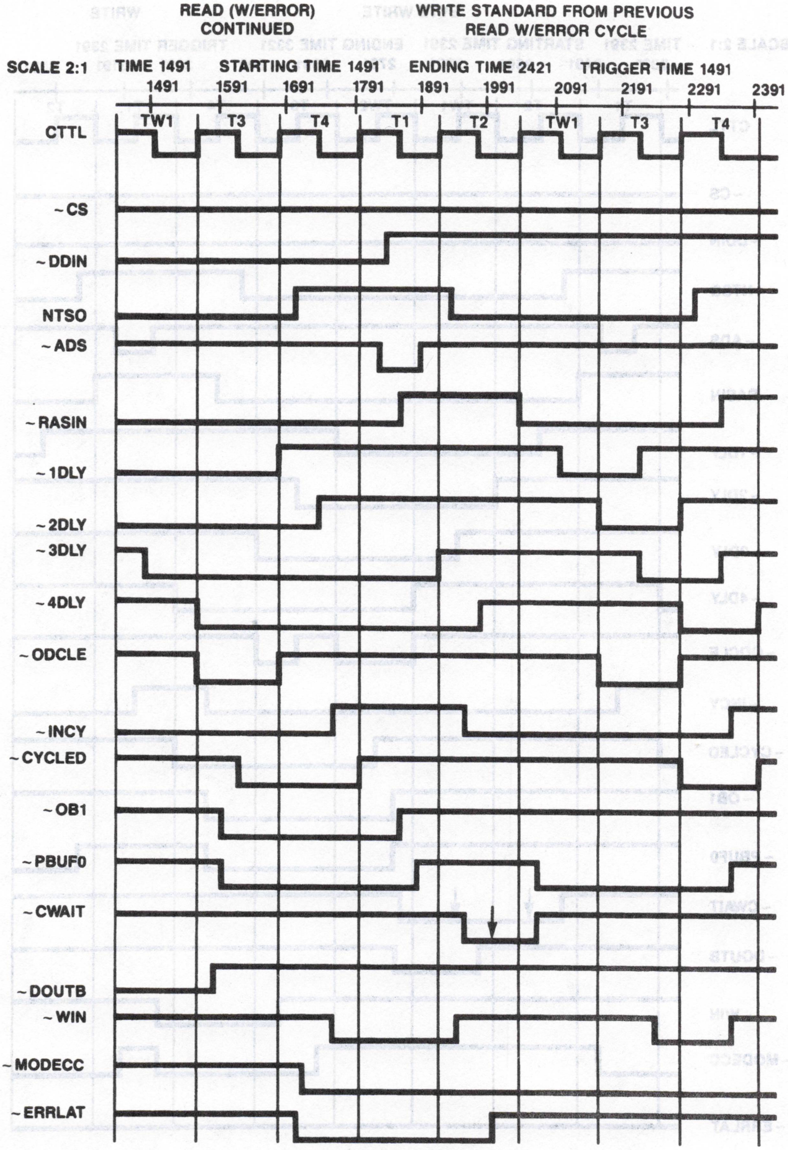

Simulation Results
READ (NO ERRORS)
READ (W/SINGLE ERROR)
SCALE 2:1
TIME 591
STARTING TIME 591
ENDING TIME 1521
TRIGGER TIME 591
591
691
791
891
991
1091
1191
1291
1391
1491
T1
T2
TW1
T3
T4
T1
T2
TW1
CTTL
~CS
~DDIN
NTSO
~ADS
~RASIN
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLE
~INCY
~CYCLED
~OB1
~PBUF0
~CWAIT
~DOUTB
~WIN
~MODECC
~ERRLAT

TL/L/9991-D4

FIGURE 7.22.8. Simulation Timing Diagram; Read/Write without Errors

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

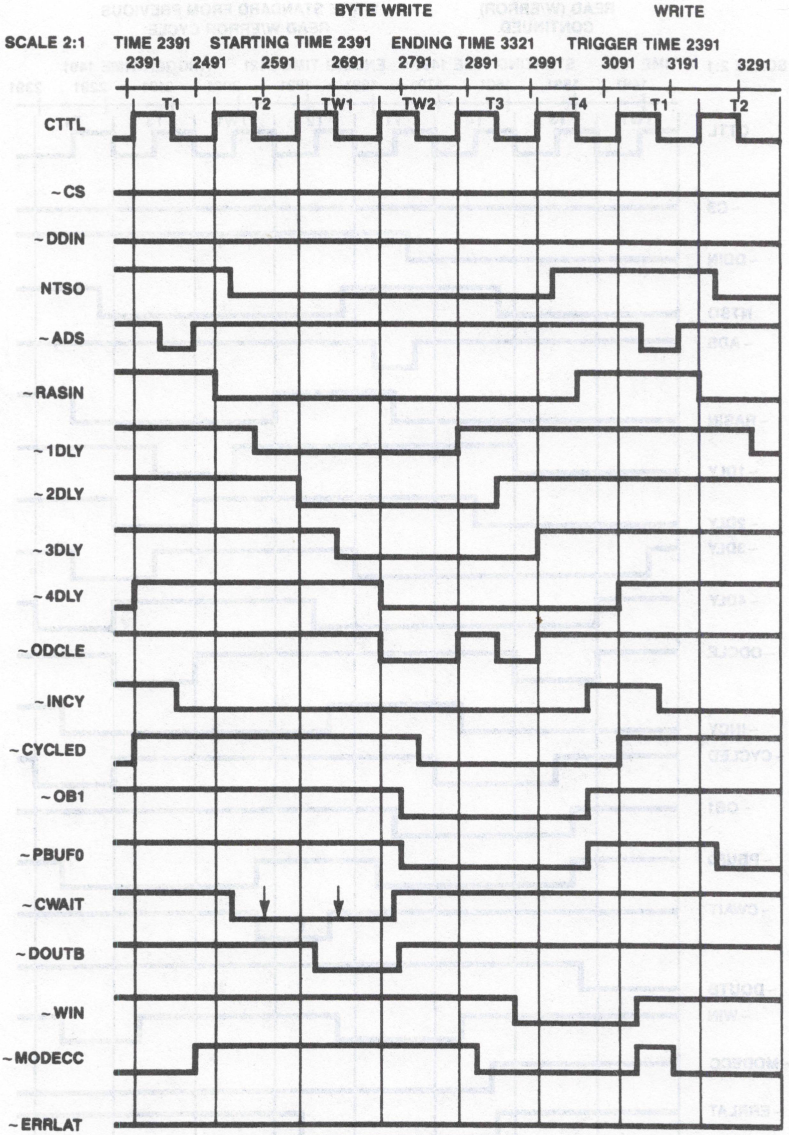

READ (W/ERROR) CONTINUED
WRITE STANDARD FROM PREVIOUS READ W/ERROR CYCLE
SCALE 2:1
TIME 1491
STARTING TIME 1491
ENDING TIME 2421
TRIGGER TIME 1491
1491
1591
1691
1791
1891
1991
2091
2191
2291
2391
TW1
T3
T4
T1
T2
TW1
T3
T4
CTTL
~CS
~DDIN
NTSO
~ADS
~RASIN
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLE
~INCY
~CYCLED
~OB1
~PBUF0
~CWAIT
~DOUTB
~WIN
~MODECC
~ERRLAT

TL/L/9991-D5

FIGURE 7.22.9. Simulation Timing Diagram; Read with Error and Write Cycle

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

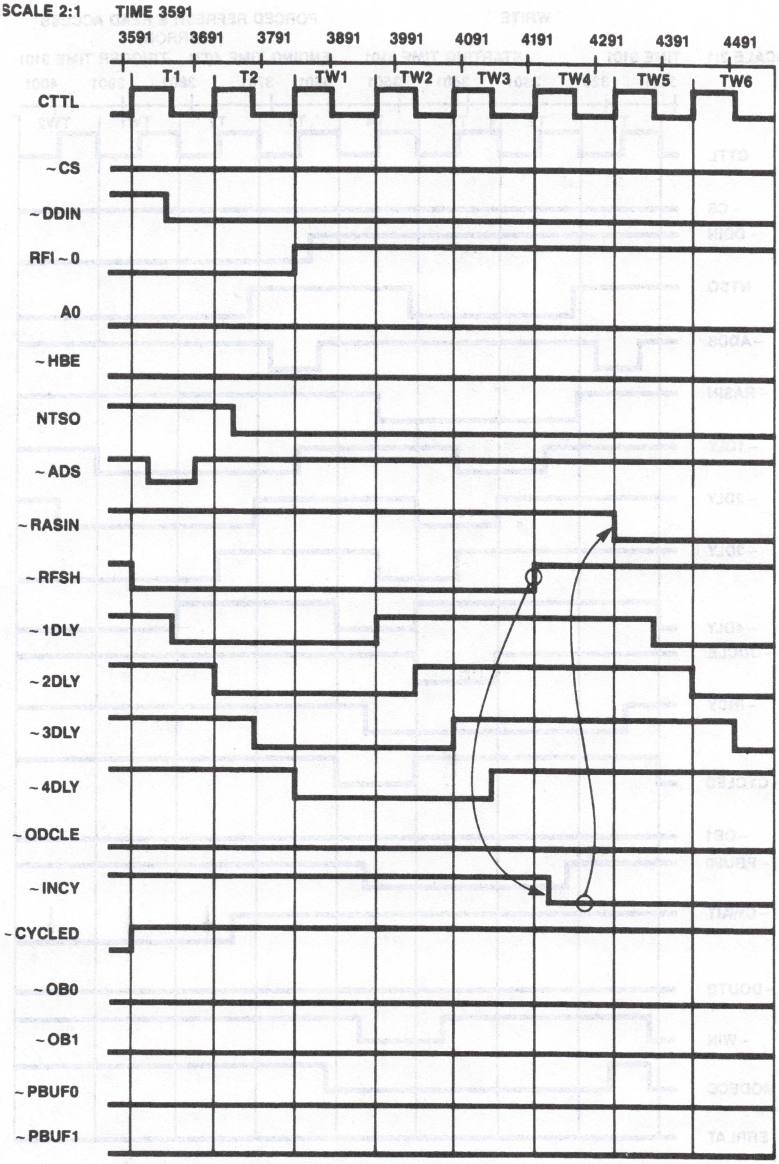

BYTE WRITE
WRITE
SCALE 2:1
TIME 2391
STARTING TIME 2391
ENDING TIME 3321
TRIGGER TIME 2391
2391
2491
2591
2691
2791
2891
2991
3091
3191
3291
T1
T2
TW1
TW2
T3
T4
T1
T2
CTTL
~CS
~DDIN
NTSO
~ADS
~RASIN
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLE
~INCY
~CYCLED
~OB1
~PBUF0
~CWAIT
~DOUTB
~WIN
~MODECC
~ERRLAT

TL/L/9991–D6

FIGURE 7.22.10. Simulation Timing Diagram; Byte Write

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

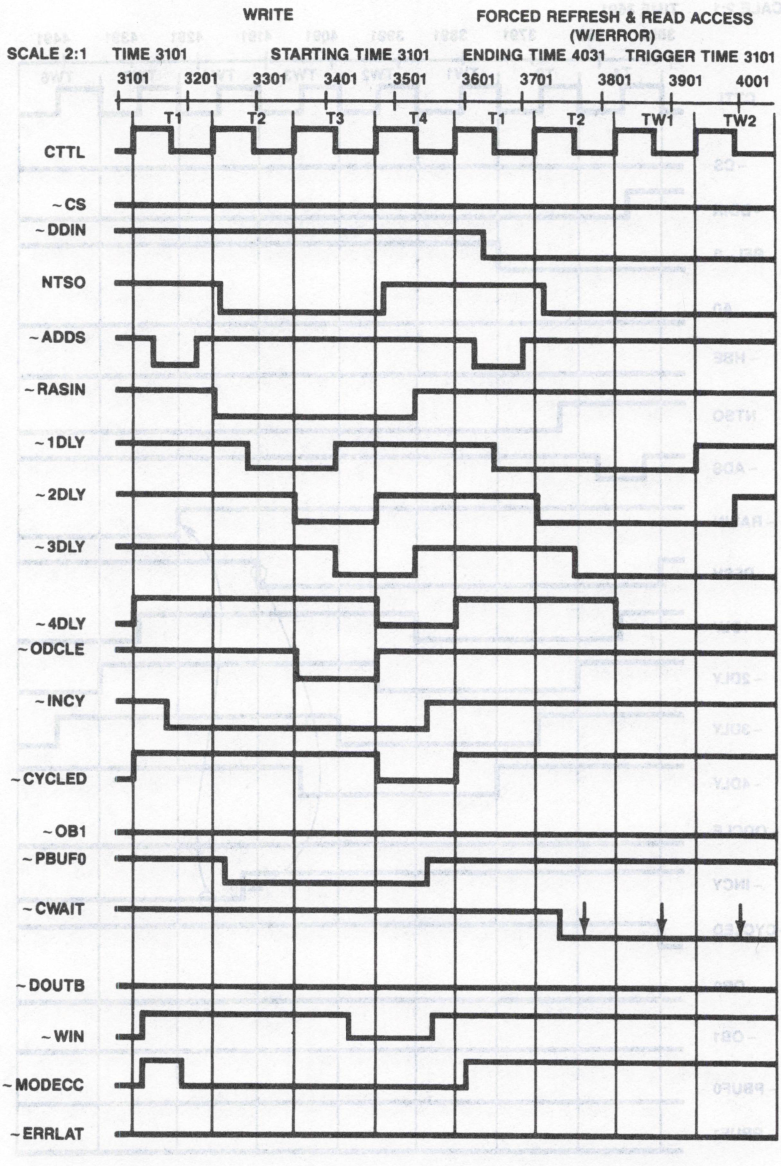

SCALE 2:1
TIME 3591
3591
3691
3791
3891
3991
4091
4191
4291
4391
4491
T1
T2
TW1
TW2
TW3
TW4
TW5
TW6
CTTL
~CS
~DDIN
RFI~0
A0
~HBE
NTSO
~ADS
~RASIN
~RFSH
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLE
~INCY
~CYCLED
~OB0
~OB1
~PBUF0
~PBUF1

TL/L/9991-D7

FIGURE 7.22.11. Simulation Timing Diagram; Forced Refresh then Access

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

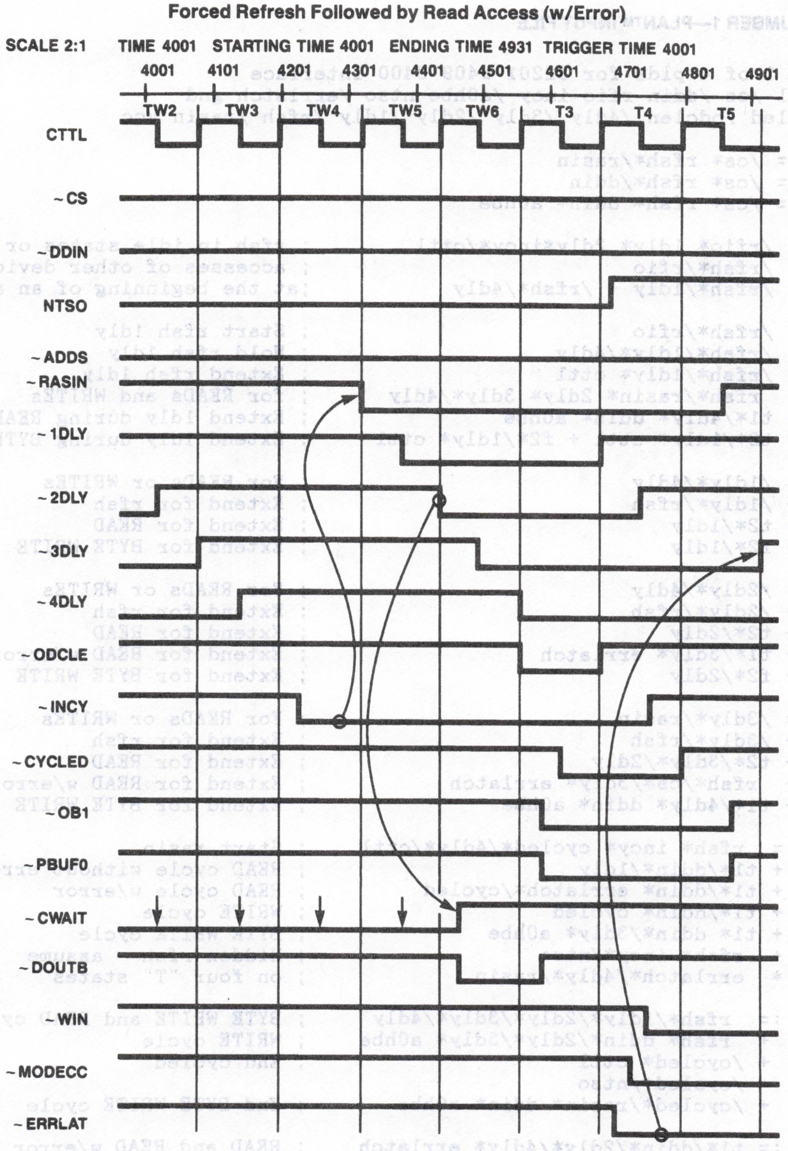

WRITE
FORCED REFRESH & READ ACCESS (W/ERROR)
SCALE 2:1
TIME 3101
STARTING TIME 3101
ENDING TIME 4031
TRIGGER TIME 3101
3101
3201
3301
3401
3501
3601
3701
3801
3901
4001
T1
T2
T3
T4
T1
T2
TW1
TW2
CTTL
~CS
~DDIN
NTSO
~ADDS
~RASIN
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLE
~INCY
~CYCLED
~OB1
~PBUF0
~CWAIT
~DOUTB
~WIN
~MODECC
~ERRLAT

TL/L/9991-D8

FIGURE 7.22.12. Simulation Timing Diagram; Write, Forced Refresh and Read Access

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)


Forced Refresh Followed by Read Access (w/Error)
SCALE 2:1
TIME 4001
STARTING TIME 4001
ENDING TIME 4931
TRIGGER TIME 4001
4001
4101
4201
4301
4401
4501
4601
4701
4801
4901
TW2
TW3
TW4
TW5
TW6
T3
T4
T5
CTTL
~CS
~DDIN
NTSO
~ADDS
~RASIN
~1DLY
~2DLY
~3DLY
~4DLY
~ODCLE
~INCY
~CYCLED
~OB1
~PBUF0
~CWAIT
~DOUTB
~WIN
~MODECC
~ERRLAT

TL/L/9991-D9

FIGURE 7.22.13. Simulation Timing Diagram; Forced Refresh Followed by Read Access (with Error)

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PAL DEVICE NUMBER 1—PLAN™ INPUT FILE

```
PAL16R8; 1 of 4 plds for 32201 8409 8400 interface
fclk cttl /cs /ddin rfio incy /a0hbe ntso /errlatch gnd
/oe /cycled /odclen /4dly /3dly /2dly /1dly /rfsh /rasin vcc

t1.term = /cs* rfsh*/rasin
t2.term = /cs* rfsh*/ddin
f2.term = /cs* rfsh* ddin* a0hbe

/rfsh := /rfio* 1dly* 2dly*incy*/cttl          ; rfsh in idle states or in long
       + /rfsh*/rfio                           ; accesses of other devices or
       + /rfsh*/1dly + /rfsh*/4dly             ;at the beginning of an access

/1dly := /rfsh*/rfio                            ; Start rfsh 1dly
       + /rfsh*/1dly*/4dly                      ; Hold rfsh 1dly
       + /rfsh*/1dly* cttl                      ; Extend rfsh 1dly
       +  rfsh*/rasin* 2dly* 3dly*/4dly         ; for READs and WRITEs
       + t1*/4dly* ddin* a0hbe                  ; Extend 1dly during READ
       + t2*/1dly* cttl + f2*/1dly* cttl        ; Extend 1dly during BYTE WRITEs

/2dly := /1dly*/4dly                            ; For READs or WRITEs
       + /1dly*/rfsh                            ; Extend for rfsh
       + t2*/1dly                               ; Extend for READ
       + f2*/1dly                               ; Extend for BYTE WRITE

/3dly := /2dly*/4dly                            ; For READs or WRITEs
       + /2dly*/rfsh                            ; Extend for rfsh
       + t2*/2dly                               ; Extend for READ
       + t1*/3dly* errlatch                     ; Extend for READ w/error
       + f2*/2dly                               ; Extend for BYTE WRITE

/4dly := /3dly*/rasin                           ; For READs or WRITEs
       + /3dly*/rfsh                            ; Extend for rfsh
       + t2*/3dly*/2dly                         ; Extend for READ
       +  rfsh*/cs*/3dly* errlatch              ; Extend for READ w/error
       + t1*/4dly* ddin* a0hbe                  ; Extend for BYTE WRITE

/rasin :=  rfsh* incy* cycled*/4dly*/cttl       ; Start rasin
        + t1*/ddin*/1dly                        ; READ cycle without error
        + t1*/ddin* errlatch*/cycled            ; READ cycle w/error
        + t1*/ddin* cycled                      ; WRITE cycle
        + t1* ddin*/3dly* a0hbe                 ; BYTE WRITE cycle
        +  rfsh* incy*/ntso                     ; Hidden rfsh - assume
        *  errlatch*/4dly*/rasin                ; on four "T" states

/cycled :=  rfsh*/1dly*/2dly*/3dly*/4dly        ; BYTE WRITE and READ cycles
         +  rfsh* ddin*/2dly*/3dly* a0hbe       ; WRITE cycle
         + /cycled* cttl                        ; End cycled
         + /cycled*/ntso
         + /cycled*/rasin* ddin* a0hbe          ; End BYTE WRITE cycle

/odclen := t1*/ddin*/2dly*/4dly* errlatch       ; READ and READ w/error
         + t1* ddin* 2dly* 3dly*/4dly* a0hbe    ; WRITE cycle
         + t1* ddin*/2dly*/4dly* a0hbe          ; BYTE WRITE
         + t1* ddin*/1dly*/cycled* a0hbe        ; BYTE WRITE
```

TL/L/9991–H3

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PAL DEVICE NUMBER 2—PLAN™ INPUT FILE

```
PAL16L8; 2 of 4 plds for 32201 8409 8400 interface
/rfsh /rasin a0 /hbe /ddin /ads ntso /2dly /4dly gnd
/errlatch cwait csoe /incy /doutb /pbuf1 /pbuf0 /ob1 /ob0 vcc

t1.term =  doutb*/ddin*/4dly*/rasin* rfsh
t2.term =  doutb* ddin*/4dly*/rasin* rfsh

/ob0 = t1                                     ; READ or READ w/error
     + t2* a0*/hbe                            ; BYTE WRITE high BYTE
ob0.trst = csoe

/ob1 = t1                                     ; READ or READ w/error
     + t2*/a0* hbe                            ; BYTE WRITE low BYTE
ob1.trst = csoe

/pbuf0 = t1*/a0                               ; READ or READ w/error
       + t2*/a0* hbe                          ; BYTE WRITE
       +  doutb*/a0*/hbe* ddin*/rasin* rfsh   ; Word WRITE
pbuf0.trst = csoe

/pbuf1 = t1*/hbe;                             ; READ or READ w/error
       + t2* a0*/hbe                          ; BYTE WRITE
       +  doutb*/a0*/hbe* ddin*/rasin* rfsh   ; Word WRITE
pbuf1.trst = csoe

/doutb = /ddin* rfsh*/2dly* 4dly              ; READ cycle
       + /a0* hbe* ddin* rfsh*/2dly* 4dly     ; BYTE WRITE
       +  a0*/hbe* ddin* rfsh*/2dly* 4dly     ; BYTE WRITE
doutb.trst = csoe

/incy =  rfsh*/ads* 4dly                      ; Start incy
       +  rfsh* csoe                          ; Start incy for access after
       * /ntso* rasin                         ; forced refresh or READ w/error
       + /incy* 4dly                          ; Continue incy
       + /incy* csoe* ddin*/rasin             ; WRITE cycles
       + /incy*/csoe*/rasin                   ; Non-/cs cycles

/cwait = /rfsh* csoe*/ntso                    ; Access in rfsh
       +  rfsh* csoe*/ntso* rasin             ; Access after forced refresh
       +  rfsh*/ddin*/rasin* 2dly*/incy* 4dly ; READ cycle
       +  rfsh* ddin*/a0* hbe*/rasin* 4dly    ; BYTE WRITE
       +  rfsh* ddin* a0*/hbe*/rasin* 4dly    ; BYTE WRITE
       + /rfsh*/incy*/errlatch* 2dly*ntso     ; Insert WAITs into next cycle
cwait.trst = csoe
```

TL/L/9991–H4

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PAL DEVICE NUMBER 3—PLAN™ INPUT FILE

```
PAL14L4; 3 of 4 plds for 32201 8409 8400 interface
/ddin /rfsh /a0hbe /errlatch /1dly /2dly /3dly /4dly /reset gnd
ae e0 e1 nc /modecc /win /parityerr /prtyr csoe vcc

/win =   rfsh*/errlatch* 2dly*/3dly*/4dly* csoe         ; READ w/error
     +   rfsh*/ddin*/3dly* a0hbe* csoe                   ; Word WRITE
     +   rfsh* ddin*/a0hbe* 2dly*/4dly* csoe             ; BYTE WRITE

/modecc =  rfsh*/errlatch* 1dly*/4dly* csoe             ; READ w/error
        +  rfsh* ddin* a0hbe* csoe                       ; Word WRITE
        +  rfsh* ddin*/a0hbe* 1dly*/4dly* csoe           ; BYTE WRITE

/parityerr =  e0*/e1*/a0hbe
           *   rfsh* ddin* reset*/4dly*/ae* csoe   ; WRITE error BYTE 1
           + /e0* e1*/a0hbe
           *   rfsh* ddin* reset*/4dly*/ae* csoe   ; WRITE error BYTE 0
           + /e0*/e1* a0hbe
           *   rfsh* ddin* reset*/4dly*/ae* csoe   ; Error both BYTEs
```

TL/L/9991–H5

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PAL DEVICE NUMBER 4 — PLAN™ INPUT FILE

```
PAL16R6; 4 of 4 plds for 32201 8409 8400 interface
fclk /odclen diagcs diagd /reset /cycled ae /e01 /3dly gnd
/oe /ddin nc /errlatch /doublerr /ole csle dle /rfsh vcc

/dle := /odclen
      +  dle* diagd                                 ; Hold /dle for diagnostics

/csle := /odclen
       +  csle* diagcs                              ; Hold /clse for diagnostics

/ole :=/odclen

/doublerr :=  rfsh*/diagcs*/diagd* reset
          * /ole*/cycled* ae* e01                   ; 2 bit error during READs
          + /doublerr* reset                        ; or BYTE WRITEs

/errlatch :=  ddin*/ole*/cycled*/diagcs*/diagd* ae ; Error during READ
          + /errlatch*/3dly
```

TL/L/9991-H6

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PLAN™ JEDEC FILE FOR PAL DEVICE #1

```
PLAN v3.12   09-06-1988    12:00
Source filename: DEV1.BEQ    Device: PAL16R8
1 of 4 plds for 32201 8409 8400 interface *
QP20* QF2048* F0*
L0000
10111101111111110111111011111101
11101001101011111111111111111111
11101001101111111111111111110110
11101001101111111111111111111101
11101001011111111110011111111111
11111101111111110111111111101111*
L0256
10111111110110010111111111111111
11111101111101111111111111111111
11111101101111111111111111111111
11111101111111111111111011111111*
L0512
11111101111101111111111111111111
11111101110111111111111011111111
01111101110111111111111111111111
11101101111111011101111011111111
11101001011111111111011011111111
01111001101011111111111111111111
01111001011011111111011111111111*
L0768
11111111111011111111111011111111
11111101110111111111111111111111
11111001101011111111111111111111
11111001011011111111011111111111*
L1024
11111111111111101111111011111111
11111101111111101111111111111111
11111001101111101111111111111111
11101001111111111110111111110111
11111001011111101111011111111111*
L1280
11101111111111111110111111111111
11111101111111111110111111111111
11111001101111101110111111111111
11111001111111111110111111110111
11101001011111111111011011111111*
L1536
11101001101111101111111011110111
11101001011111011101011011111111
11101001011111101111011011111111
11101001011011111111011111111110*
L1792
11111101111011101110111011111111
11111101011111101110011111111111
01111111111111111111111111111110
11111111111111111111111110111110
11101110111111111111011111111110*
C81AA* 2912
```

TL/L/9991-H7

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PLAN™ JEDEC FILE FOR PAL DEVICE #2

```
PLAN v3.12   09-06-1988   11:42
Source filename: DEV2.BEQ   Device: PAL16L8
2 of 4 plds for 32201 8409 8400 interface *
QP20* QF2048* F0*
L0000
11111111111111111111111111011111
10011111111110111101111111111011
10010111101101111101111111111011*
L0256
11111111111111111111111111011111
10011111111110111101111111111011
10011011011101111101111111111011*
L0512
11111111111111111111111111011111
10011011111110111101111111111011
10011011011101111101111111111011
10011011101101111101111111111111*
L0768
11111111111111111111111111011111
10011111101110111101111111111011
10010111101101111101111111111011
10011011101101111101111111111111*
L1024
11111111111111111111111111111111
11011111111110111111111110110111
11011011011101111111111110110111
11010111101101111111111110110111*
L1280
11111111111111111111111111111111
11011111111111111011111111110111
01011111111111111111101111011111*
L1408
11111111111111111111111011110111
10111111111101111111111011011111
10111111111111111111111011101111*
L1792
11111111111111111111111111011111
11101111111111111111101111011111
01011111111111111111101111011111
10101111111101111111111010110111
10101011011101111111111111110111
10100111101101111111111111110111
11011111111111111111011001111110*
C730F* F13E
```

TL/L/9991-H8

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

PLAN™ JEDEC FILE FOR PAL DEVICE #3

```
PLAN v3.12   09-06-1988   12:37
Source filename: DEV3.BEQ   Device: PAL14L4
3 of 4 plds for 32201 8409 8400 interface *
QP20* QF448* F0*
L000
0101100111111111111010010110
0101100111111111110110100110
0101010111111111111010100110*
L112
0111110110111101101110111111
0110010111111111110111111111
0101100111111101111110111111*
L224
0111110110110111111110111111
0101010111111111111111111111
0101100111110111111110111111*
C1787* 59AF
```

TL/L/9991-H9

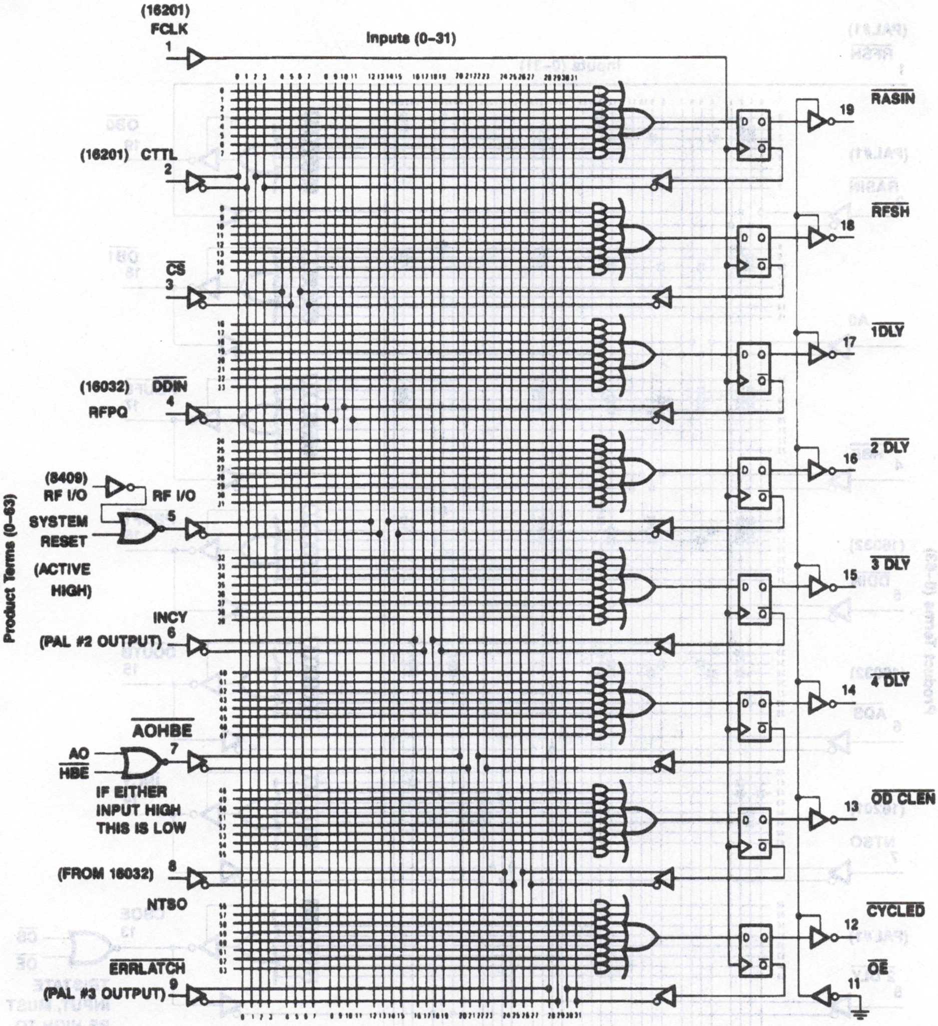
PLAN™ JEDEC FILE FOR PAL DEVICE #4

```
PLAN v3.12   09-06-1988   12:31
Source filename: DEV4.BEQ   Device: PAL16R6
4 of 4 plds for 32201 8409 8400 interface *
QP20* QF2048* F0*
L0256
10111111111111111111111111111111
11111010111111111111111111111111*
L0512
10111111111111111111111111111111
11110111110111111111111111111111*
L0768
10111111111111111111111111111111*
L1024
11011011101101101011011101111111
11111111111101111110111111111111*
L1280
11111011101111110101101111111101
11111111111111111111111011111011*
C210C* 6482
```

TL/L/9991-I0

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)


(16201)
FCLK
1
Inputs (0–31)
(16201) CTTL
2
CS
3
(16032) DDIN
4
RFPQ
(8409)
RF I/O
RF I/O
SYSTEM
RESET
(ACTIVE
HIGH)
5
Product Terms (0–63)
INCY
(PAL #2 OUTPUT)
6
AOHBE
AO
HBE
7
IF EITHER
INPUT HIGH
THIS IS LOW
(FROM 16032)
8
NTSO
ERRLATCH
(PAL #3 OUTPUT)
9
RASIN
19
RFSH
18
1DLY
17
2 DLY
16
3 DLY
15
4 DLY
14
OD CLEN
13
CYCLED
12
OE
11
D
Q
Q̄

TL/L/9991–E6

FIGURE 7.22.14. PAL16R8 Logic Diagram Showing Pinout of PAL Device # 1

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

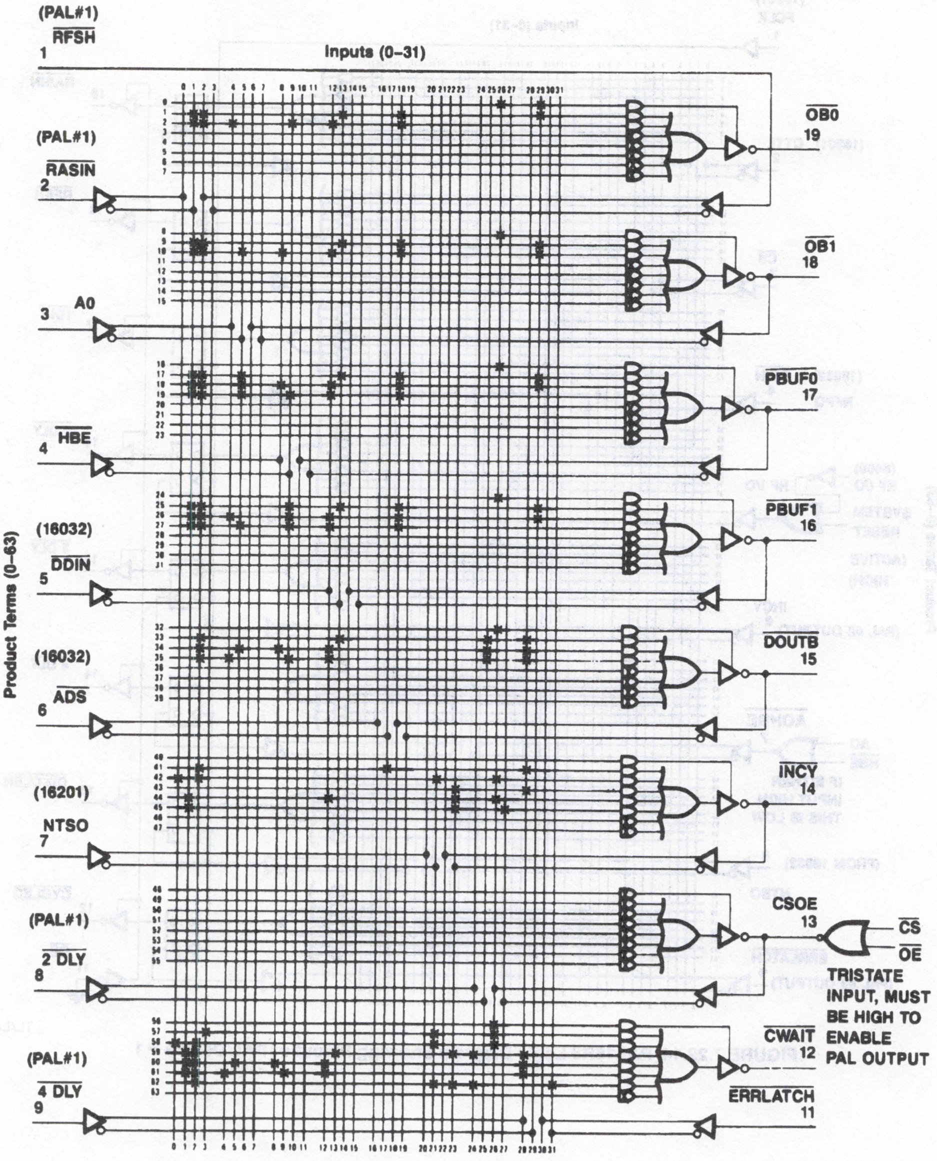
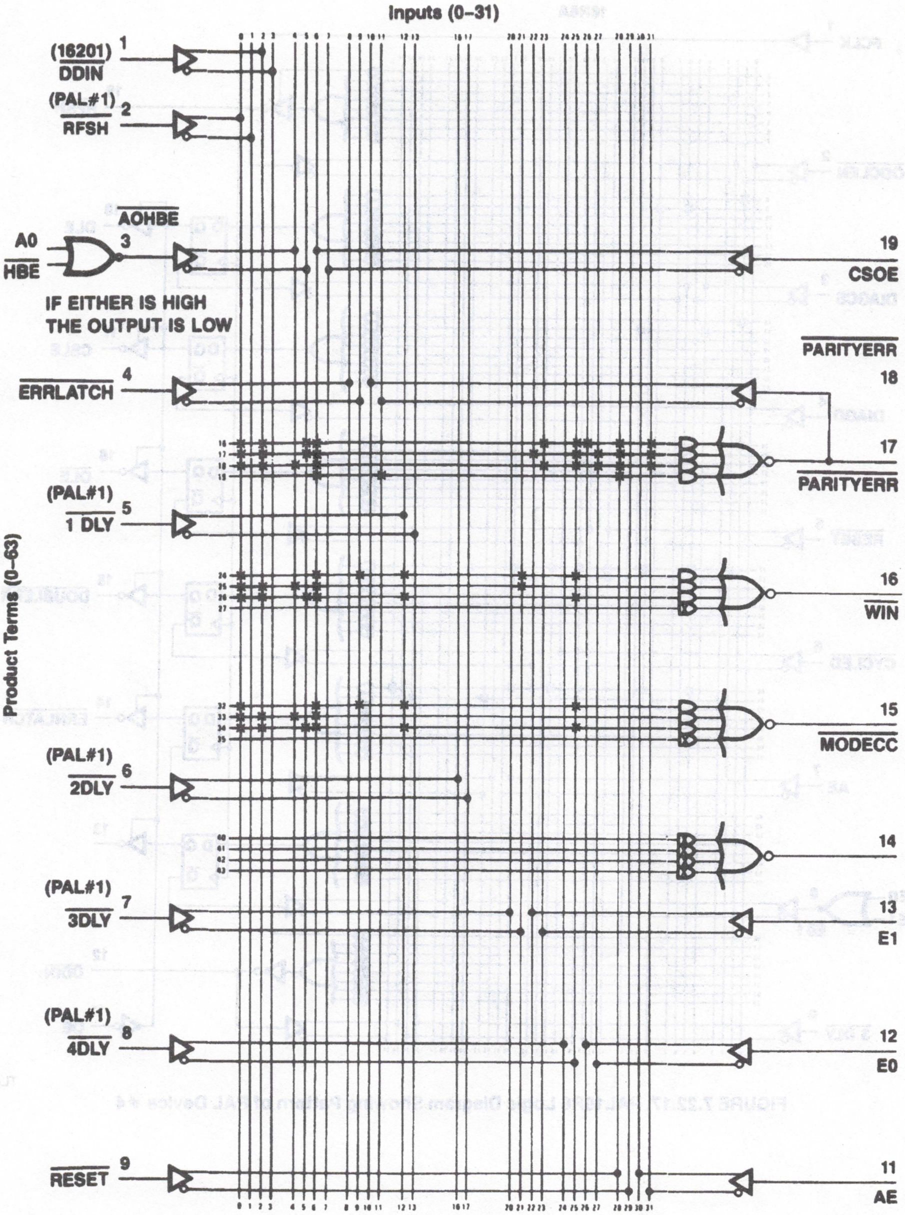

(PAL#1)
RFSH
1
Inputs (0–31)
(PAL#1)
RASIN
2
A0
3
HBE
4
(16032)
DDIN
5
(16032)
ADS
6
(16201)
NTSO
7
(PAL#1)
2 DLY
8
(PAL#1)
4 DLY
9
Product Terms (0–63)
OB0
19
OB1
18
PBUF0
17
PBUF1
16
DOUTB
15
INCY
14
CSOE
13
CS
OE
TRISTATE INPUT, MUST BE HIGH TO ENABLE PAL OUTPUT
CWAIT
12
ERRLATCH
11
TL/L/9991-E7

FIGURE 7.22.15. PAL16L8 Logic Diagram Showing Pattern of PAL Device #2

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)

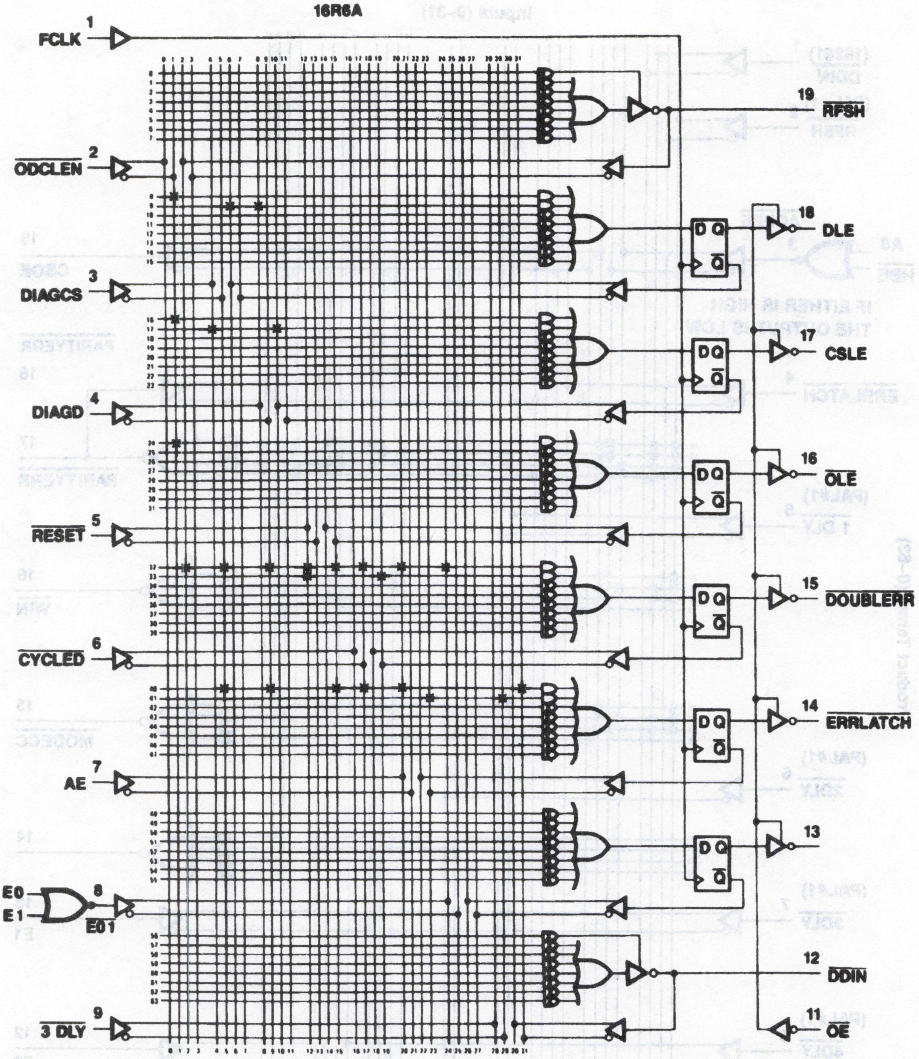

Inputs (0–31)
(16201)
DDIN
1
(PAL#1)
RFSH
2
AOHBE
A0
HBE
3
IF EITHER IS HIGH
THE OUTPUT IS LOW
ERRLATCH
4
(PAL#1)
1 DLY
5
(PAL#1)
2DLY
6
(PAL#1)
3DLY
7
(PAL#1)
4DLY
8
RESET
9
Product Terms (0–63)
19
CSOE
PARITYERR
18
17
PARITYERR
16
WIN
15
MODECC
14
13
E1
12
E0
11
AE

TL/L/9991-E8

FIGURE 7.22.16. PAL14L4 Logic Diagram Showing Pattern of PAL Device #3

7.22 DP8409 DRAM Controller and DP8400 ECC Interface for NS32032

(Continued)


16R6A
FCLK
1
ODCLEN
2
DIAGCS
3
DIAGD
4
RESET
5
CYCLED
6
AE
7
E0
E1
8
E01
3 DLY
9
19
RFSH
18
DLE
17
CSLE
16
OLE
15
DOUBLERR
14
ERRLATCH
13
12
DDIN
11
OE
D Q
Q

TL/L/9991-E9

FIGURE 7.22.17. PAL16R6 Logic Diagram Showing Pattern of PAL Device #4

Section 8

Appendices/ Physical Dimensions

Section 8 Contents

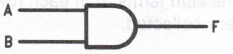
Appendix A—Boolean Logic Review 8-3
  A.1 Basic Operators and Theorems 8-3
  A.2 Derivation of a Boolean Expression 8-4
  A.3 Minimization 8-6
  A.4 K-Map Method 8-7
  A.5 Sequential Circuit Elements 8-11
  A.6 State Machine Fundamentals 8-12
  A.7 Avoiding Logic Hazards 8-14
Appendix B—Theory of PLD Testing 8-17
  B.1 Testing Methods 8-17
  B.2 Test Generation Techniques 8-18
  B.3 Response Evaluation Techniques 8-28
Appendix C—Physical Dimensions 8-33

Bookshelf
Distributors

Appendix A
Boolean Logic Review

A.1 Basic Operators and Theorems

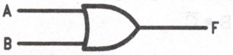
A gate is an electronic circuit which operates on one or more input signals to produce an output signal. There are three basic gates from which all other logic can be realized: AND, OR, and INVERTER gates. *Figure A.1.1* shows these three basic gates and their truth table.



TL/L/9992-7

(A) AND Gate

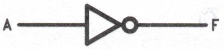
Input		Output
A	**B**	**F**
0	0	0
0	1	0
1	0	0
1	1	1



TL/L/9992-8

(B) OR Gate

Input		Output
A	**B**	**F**
0	0	0
0	1	1
1	0	1
1	1	1



TL/L/9992-9

(C) Inverter

Input	Output
A	**F**
0	1
1	0

FIGURE A.1.1. Basic Gates

To express the function of these gates by Boolean algebra, we need to define Boolean operators as follows:

$=$ Logical Equality

$\overline{x}$ Negate (Not, Invert, Complement)

$+$ OR (Sum)

$\bullet$ AND (Product)

$\oplus$ Exclusive OR

The function of an AND gate in *Figure A.1.1* can be expressed as:

$$F = A \bullet B$$

The function of an OR gate and INVERTER can be expressed as:

$$F = A + B$$

and $F = \overline{A}$

Boolean operators are logical operators, which are different from arithmetic operators. For example, + is logical addition, • is logical multiplication. We call such equations Boolean equations or logic equations.

A number of logic theorems and laws will be used to manipulate and reduce logical equations. These theorems and laws are as follows:

Theorem 1 $A + 0 = A$

Theorem 2 $A \bullet 0 = 0$

Theorem 3 $A + 1 = 1$

Theorem 4 $A \bullet 1 = A$

Theorem 5 $A + A = A$

Theorem 6 $A \bullet A = A$

Theorem 7 $A + \overline{A} = 1$

Theorem 8 $A \bullet \overline{A} = 0$

Theorem 9 $\overline{\overline{A}} = A$

Theorem 10 $A + A \bullet B = A$

Theorem 11 $A \bullet (A + B) = A$

Theorem 12 $(A + B) \bullet (A + C) = A + B \bullet C$

Theorem 13 $A + \overline{A} \bullet B = A + B$

Commutative Law

$$A + B = B + A$$
$$A \bullet B = B \bullet A$$

Associative Law

$$A + B + C = (A + B) + C = A + (B + C)$$
$$A \bullet B \bullet C = (A \bullet B) \bullet C = A \bullet (B \bullet C)$$

Distributive Law

$$A + (B \bullet C \bullet D) = (A + B) \bullet (A + C) \bullet (A + D)$$
$$A \bullet (B + C + D) = A \bullet B + A \bullet C + A \bullet D$$

DeMorgan's Theorem

$$\overline{(A + B + C)} = \overline{A} \bullet \overline{B} \bullet \overline{C}$$
$$\overline{(A \bullet B \bullet C)} = \overline{A} + \overline{B} + \overline{C}$$

The complement of any Boolean expression, or a part of any expression, may be found by means of DeMorgan's theorem. Two steps are used to form a complement in this theorem:

1. OR symbols are replaced with AND symbols or AND symbols with OR symbols.
2. Each of the terms in the expression is complemented.

DeMorgan's theorem is one of the most powerful tools for engineering applications. It is very useful for designing with programmable logic devices because it provides a quick and simple conversion method between PRODUCT-OF-SUMS and SUM-OF-PRODUCTS expressions, which will be defined later.

A.2 Derivation of a Boolean Expression

Any logic expression can be reduced to a two-level form and expressed as either a SUM-OF-PRODUCTS (SOP) or PRODUCT-OF-SUMS (POS). Before we define SOP or POS, we need to define "terms".

1. **Product Term:** A product term is a single variable or the logical product of several variables. The variable may or may not be complemented.
2. **Sum Term:** A sum term is a single variable or the sum of several variables. The variables may or may not be complemented.
3. **Normal Term:** A normal term is a product or sum term in which no variable appears more than once.
4. **Minterm:** A minterm is a product term containing every variable once and only once (either true or complemented).
5. **Maxterm:** A maxterm is a sum term containing every variable once and only once (either true or complemented).

For example, the term $A \bullet B \bullet C$ is a product term; $A + B$ is a sum term; A is both a product term and a sum term; $A + B \bullet C$ is neither a product term nor a sum term; $A + \overline{B}$ is a sum term; $A \bullet \overline{B} \bullet \overline{C}$ is a product term; $\overline{B}$ is both a sum term and a product term. We now define two most important forms:

1. **SUM-OF-PRODUCTS Expression:** A sum-of-products expression is a product term or several product terms logically added together.
2. **PRODUCT-OF-SUMS Expression:** A product-of-sums expression is a sum term or several sum terms logically multiplied together.

For example, the expression $\overline{A} \bullet B + A \bullet \overline{B}$ is a sum-of-products expression; $(A + B) \bullet (\overline{A} + \overline{B})$ is a product-of-sums expression.

One prime reason for using sum-of-products or product-of-sums expressions is their straightforward conversion to very simple gating networks. In their purest, simplest form they go into two-level networks, which are networks for which the longest path through which a signal must pass from input to output is two gates long.

When designing a logic circuit, the logic designer works from two sets of known values; the various states which the inputs to the logical network can take, and the desired outputs for each input condition. The logic expression is derived from these sets of values and the procedure is as follows:

1. Construct a table of the input and output values (Table A.2.1 left half).

2a. To derive a SUM-OF-PRODUCTS (SOP) expression:

A product term column is added listing the inputs A, B, and C according to their value in the input columns (Table A.2.1). Then the product terms from each row in which the output is a "1" are collected.

Therefore:

$$F = \overline{A} \bullet B \bullet \overline{C} + \overline{A} \bullet B \bullet C + A \bullet B \bullet \overline{C} \qquad \text{(Eq. A.2.1)}$$

2b. To derive a PRODUCT-OF-SUMS (POS) expression:

A sum term column is added listing the inputs A, B, and C according to their *complement* value in the input columns (Table A.2.1). Then the sum terms from each row in which the output is "0" are collected.

Therefore:

$$F = (A + B + C)\,(A + B + \overline{C})\,(\overline{A} + B + C)$$
$$(\overline{A} + B + \overline{C})\,(\overline{A} + \overline{B} + \overline{C}) \qquad \text{(Eq. A.2.2)}$$

Figure A.2.1 is the logic circuit derived from Eq A.2.1 *Figure A.2.2* is derived from Eq. A.2.2.

Eq. A.2.1 Can be simplified as shown below:

$$\begin{aligned}
F &= \overline{A} \bullet B \bullet \overline{C} + \overline{A} \bullet B \bullet C + A \bullet B \bullet \overline{C} \\
&= \overline{A} \bullet B\,(\overline{C} + C) + A \bullet B \bullet \overline{C} \\
&= \overline{A} \bullet B + A \bullet B \bullet \overline{C} \\
&= B\,(\overline{A} + A \bullet \overline{C}) \\
&= B\,(\overline{A} + \overline{C}) \\
&= \overline{A} \bullet B + B \bullet \overline{C}
\end{aligned}$$

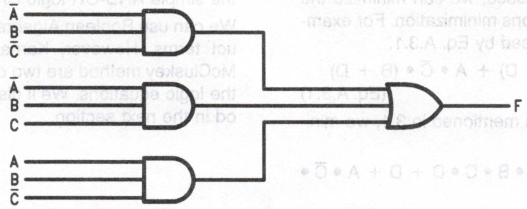
Eq. A.2.2 can be simplified as shown:

$$\begin{aligned}
F = {} & (A + B + C)\,(A + B + \overline{C})\,(\overline{A} + B + C) \\
& (\overline{A} + B + \overline{C})\,(\overline{A} + \overline{B} + \overline{C}) \\
= {} & (A + B)\,(\overline{A} + B)\,(\overline{A} + \overline{C}) \\
= {} & B\,(\overline{A} + \overline{C}) \\
= {} & \overline{A} \bullet B + B \bullet \overline{C}
\end{aligned}$$

TABLE A.2.1. Truth Table Eq. A.2.1 and Eq. A.2.2

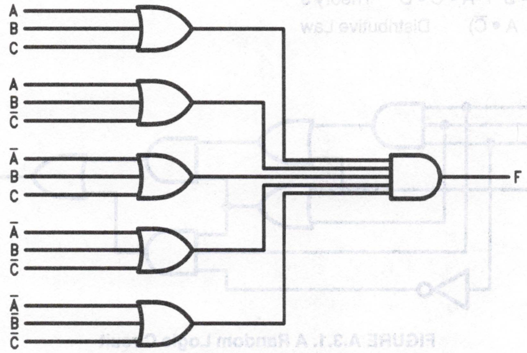
Inputs			Outputs	Product Terms	Sum Terms
A	B	C	F		
0	0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$	$A + B + C$
0	0	1	0	$\overline{A}\,\overline{B}\,C$	$A + B + \overline{C}$
0	1	0	1	$\overline{A}\,B\,\overline{C}$	$A + \overline{B} + C$
0	1	1	1	$\overline{A}\,B\,C$	$A + \overline{B} + \overline{C}$
1	0	0	0	$A\,\overline{B}\,\overline{C}$	$\overline{A} + B + C$
1	0	1	0	$A\,\overline{B}\,C$	$\overline{A} + B + \overline{C}$
1	1	0	1	$A\,B\,\overline{C}$	$\overline{A} + \overline{B} + C$
1	1	1	1	$A\,B\,C$	$\overline{A} + \overline{B} + \overline{C}$

The two final expressions obtained are identical and can be implemented by the circuit shown in *Figure A.2.3.* This is much simpler than the circuits in *Figures A.2.1* and *A.2.2.* This simplified procedure is called minimization.


$\bar{A}$
B
$\bar{C}$
$\bar{A}$
B
C
A
B
$\bar{C}$
F

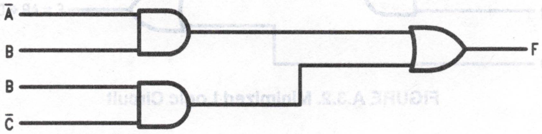
TL/L/9992-10

FIGURE A.2.1. Logic Circuits of Eq. A.2.1


A
B
C
A
B
$\bar{C}$
$\bar{A}$
B
C
$\bar{A}$
B
$\bar{C}$
$\bar{A}$
$\bar{B}$
$\bar{C}$
F

TL/L/9992-11

FIGURE A.2.2. Logic Circuits of Eq. A.2.2


$\bar{A}$
B
B
$\bar{C}$
F

TL/L/9992-12

FIGURE A.2.3. Simplified Logic Circuits

A.3 Minimization

Logic circuits can be represented by logic expressions or so called logic equations. As discussed, we can minimize the logic circuit through logic equations minimization. For example, *Figure A.3.1* can be expressed by Eq. A.3.1.

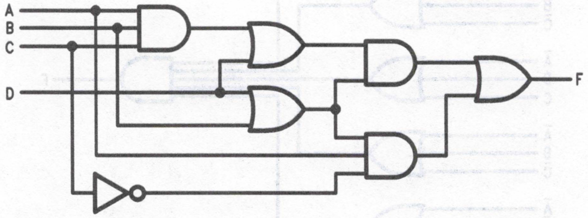
$$F = (A \bullet B \bullet C + D) \bullet (B + D) + A \bullet \overline{C} \bullet (B + D) \qquad \text{(Eq. A.3.1)}$$

By using the theorems and laws mentioned in 3.1, we minimize Eq. A.3.1 as follows:

$$\begin{aligned} F &= A \bullet B \bullet C + B \bullet D + A \bullet B \bullet C \bullet D + D + A \bullet \overline{C} \bullet B + A \bullet \overline{C} \bullet D \\ &= A \bullet B \bullet C\,(1 + D) + D(B + 1) + A \bullet \overline{C} \bullet B + A \bullet \overline{C} \bullet D && \text{Distributive Law} \\ &= A \bullet B \bullet C + D + A \bullet \overline{C} \bullet B + A \bullet \overline{C} \bullet D && \text{Theory 3} \\ &= A \bullet B\,(C + \overline{C}) + D\,(1 + A \bullet \overline{C}) && \text{Distributive Law} \\ &= A \bullet B + D \end{aligned}$$

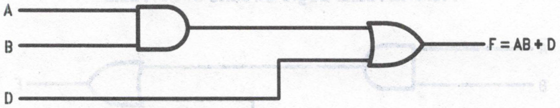
The minimum SOP expression can now be implemented as the simple AND-OR logic circuits as shown in *Figure A.3.2.*

We can use Boolean Algebra to reduce the number of product terms. However, Karnaugh Mapping and the Quine-McCluskey method are two other powerful tools to minimize the logic equations. We'll discuss Karnaugh Mapping method in the next section.



TL/L/9992-13

FIGURE A.3.1. A Random Logic Circuit



TL/L/9992-14

FIGURE A.3.2. Minimized Logic Circuit

A.4 K-Map Method

A Karnaugh map, hereafter called a K-map, is a graphical method for representing a Boolean function. It is similar to a truth table in that the K-map supplies the TRUE or FALSE value of a Boolean function for all possible combinations of its logical argument. There are many ways in which a K-map can be arranged. The most important considerations of the arrangement are:

1. There must be a unique location on the K-map for entering the TRUE/FALSE value of the function that corresponds to each combination of input variables.
2. The locations should be arranged so, with minimization mentioned in Section A.3, that they are readily apparent to the trained observer.

The second consideration implies that a successful K-mapping arrangement should point to groups of minterms or maxterms that can be combined into reduced forms. K-maps are also useful in expanding partially reduced expressions into standard forms prior to the minimization process.

The K-map is one of the most powerful tools at the hands of the logic designer. The power of the K-map does not lie in its application of any marvelous new theorems, but rather in its utilization of the remarkable ability of the human mind to perceive patterns in pictorial representations of data. This is not a new idea. Anytime we use a graph instead of a table of numerical data, we are utilizing the human ability to recognize complex patterns and relationships in a graphical representation far more rapidly and surely than in a tabular representation. A few examples of how to create a K-map follow.

First, consider a truth table for two variables. We list all four possible input combinations and the corresponding function values, i.e., the truth tables for AND and OR. *(Figure A.4.1)*

A	B	A • B
0	0	0
0	1	0
1	1	1
1	0	0

A	B	A + B
0	0	0
0	1	1
1	1	1
1	0	1

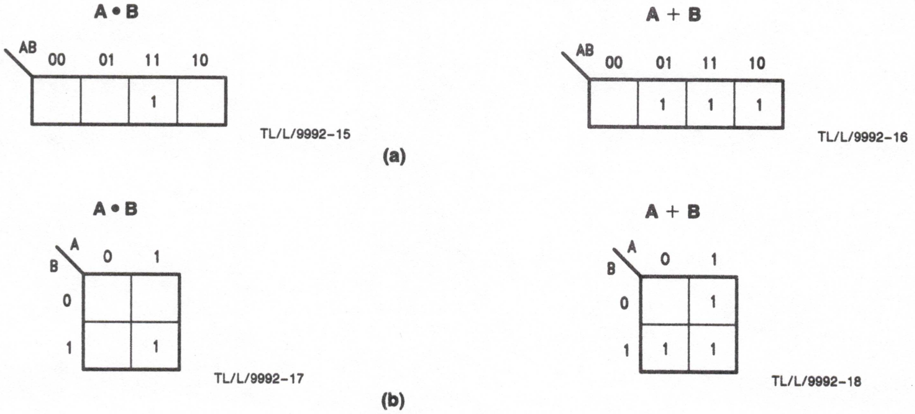
FIGURE A.4.1. Truth Tables for AND and OR

As an alternative approach, set up a diagram consisting of four small boxes, one for each combination of variables. Place a "1" in any box representing a combination of variables for which the function has the value 1. There is no logical objection to putting "0's" in the other boxes, but they are usually omitted for clarity.

The diagrams in *Figure A.4.2(a)* are perfectly valid K-maps, but it is more common to arrange the four boxes in a square, as shown in *Figure A.4.2(b)*.

Since there must be one square for each input combination, there must be 2^n squares in a K-map for n-variables. Whatever the number of variables, we may interpret the map in terms of a graphical form of the truth table *(Figure A.4.3(a))* or in terms of union and intersection of areas *(Figure A.4.3(b))*. The K-maps for some other three-variable functions are shown in *Figure A.4.4*.

Particularly note the functions mapped in *Figure A.4.3(a)* and *A.4.4(b)*. These are both minterms. Each is represented by one square, obviously, and each one of the eight squares corresponds to one of the eight minterms of three variables. This is the origin of the name minterm. A minterm is the form of Boolean function corresponding to the minimum possible area, other than 0, on a K-map. A maxterm, on the other hand, is the form of Boolean function corresponding to the maximum possible area, other than 1, on a K-map. *Figure A.4.3(b)* and *A.4.4(c)* are two examples.



TL/L/9992-15

TL/L/9992-16

(a)

TL/L/9992-17

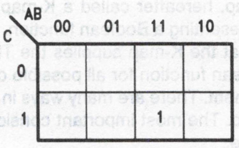
TL/L/9992-18

(b)

FIGURE A.4.2. K-Maps for AND and OR

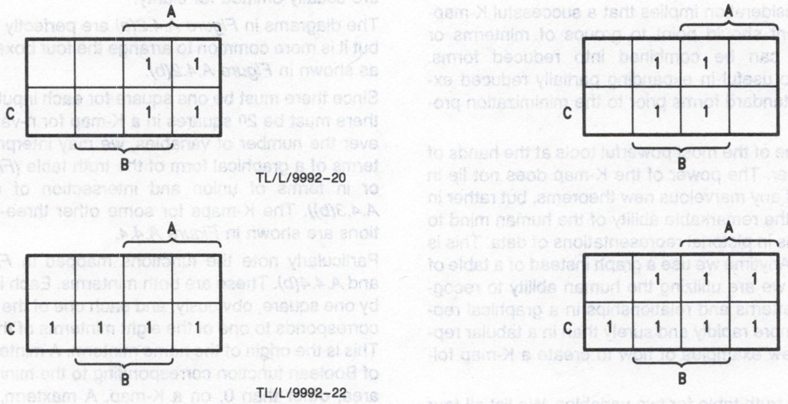
A.4 K-Map Method (Continued)

A	B	C	A•B•C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1


AB
C
00
01
11
10
0
1
1

TL/L/9992-19

(a)


A
1
1
C
1
1
B
A
1
1
C
1
1
B
A
C
1
1
1
1
B
A
1
1
1
C
1
1
1
1
B

TL/L/9992-20

TL/L/9992-21

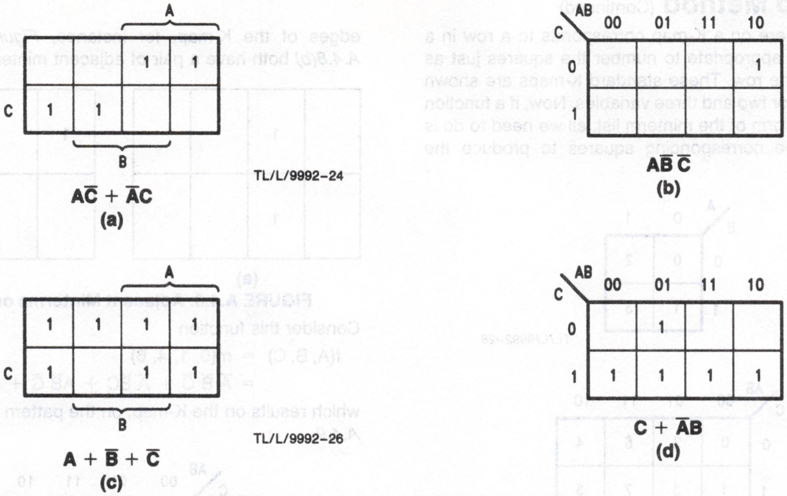
TL/L/9992-22

TL/L/9992-23

A + B + C = A+B+C

(b)

FIGURE A.4.3. K-Maps for 3-Variable AND and OR


A
C
1
1
1
1
B
TL/L/9992-24
AB
C
00
01
11
10
0
1
1
TL/L/9992-25
A
1
1
1
1
C
1
1
1
B
TL/L/9992-26
AB
C
00
01
11
10
0
1
1
1
1
1
1
TL/L/9992-27

$A\overline{C} + \overline{A}C$

(a)

$A\overline{B}\,\overline{C}$

(b)

$A + \overline{B} + \overline{C}$

(c)

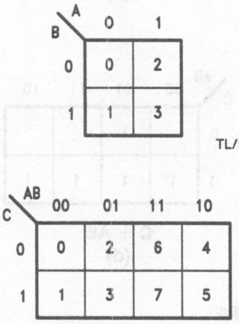
$C + \overline{A}B$

(d)

FIGURE A.4.4. Sample 3-Variable K-Maps

A.4 K-Map Method (Continued)

Since each square on a K-map corresponds to a row in a truth table, it is appropriate to number the squares just as we numbered the row. These standard K-maps are shown in *Figure A.4.5* for two and three variables. Now, if a function is stated in the form of the minterm list, all we need to do is enter 1's in the corresponding squares to produce the K-map.



TL/L/9992-28

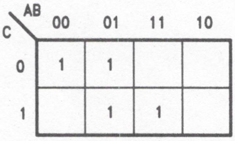
TL/L/9992-29

FIGURE A.4.5. K-Maps for Two and Three Variables

If a function is stated as a maxterm list, we can enter 0's in the squares listed or 1's in those not listed.

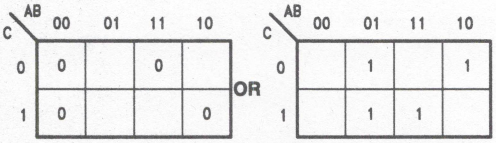
A map showing the 0's of a function is a perfectly valid K-map, although it is more common to show the 1's.

For example, the K-map of f(A, B, C) = m(0, 2, 3, 7) is shown in *Figure A.4.6* and the K-map of f(A, B, C) = M(0, 1, 5, 6) is shown in *Figure A.4.7* where m means minterm, M means maxterm.



TL/L/9992-30

FIGURE A.4.6. K-Map of M(0, 2, 3, 7)

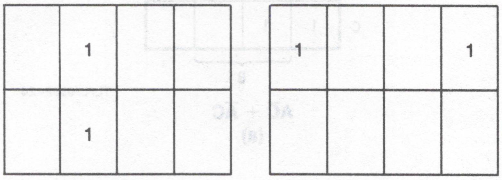


TL/L/9992-31 TL/L/9992-32

FIGURE A.4.7. K-Map of M(0, 1, 5, 6)

As shown, the K-map can be generated from the truth table on minterm expression or maxterm expression. For the remainder of this section, we will learn how to minimize the minterm expression by using the K-map.

The general principle of this minimization technique is "Any pair of n-variable minterms which are adjacent on a K-map may be combined into a single product term of n − 1 literals." The definition of "adjacent" should include opposite edges of the K-map, for instance, *Figure A.4.8(a)* and *A.4.8(b)* both have a pair of adjacent minterms.



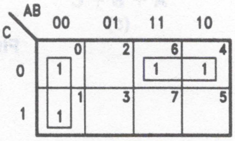
(a) (b)

FIGURE A.4.8. Adjacent Minterms on a K-Map

Consider this function

$$f(A, B, C) = m(0, 1, 4, 6)$$
$$= \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}C + A\bar{B}\,\bar{C} + AB\bar{C}$$

which results on the K-map, on the pattern shown in *Figure A.4.9*.



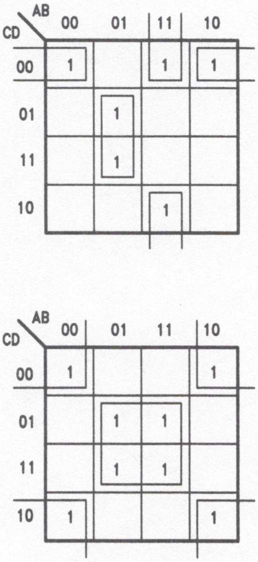
TL/L/9992-33

FIGURE A.4.9. Minimization

Therefore, combine minterms 0 and 1, 4 and 6 to get a minimal expression:

$$f(A, B, C) = \bar{A}\,\bar{B} + A\bar{C}$$

Figure A.4.10 shows some examples. Notice that it is permissible to include a minterm in several terms if it helps make the term shorter.



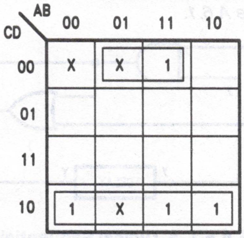
TL/L/9992-34

TL/L/9992-35

FIGURE A.4.10. Minimization

A.4 K-Map Method (Continued)

Quite often, some of the possible combinations of input values never occur. In this case, we "don't care" what the function does if these input combinations appear. The K-map makes it easy to take advantage of these "don't care" conditions by letting the "don't care" minterms be 1 or 0, depending on which value results in a simpler expression. *Figure A.4.11* shows an example of the use of "don't cares" (redundancies) to simplify the terms.



TL/L/9992–36

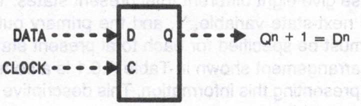
FIGURE A.4.11. Minimization

When working with larger functions, the tabular reduction developed by Quine and modified by McCluskey is an alternative to the K-map method. The Quine-McCluskey minimization method involves simple, repetitive operations that compare each minterm that is present in a sum-of-minterms expression for a Boolean functions to all other minterms with which it may form a combinable grouping.

The reader can refer to "Introduction to Switching Theory and Logic Design" by Hill and Peterson to understand the Quine-McCluskey method.

A.5 Sequential Circuit Elements

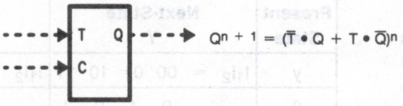
Usually the subject of logic design is subdivided into two types: sequential and combinational. A purely combinational logic subsystem has no memory. Its outputs are completely defined by its present inputs. The analysis and design of combinational logic is much easier. A sequential logic subsystem has memory and its outputs are functions of not only present inputs but the previous outputs. Circuits of multiplexer/selector, decoder/encoder, adder, and comparator are examples of combinational circuits. Shift register, counter, state machine, and memory controller are examples of sequential circuits.



$Q^{n+1} = D^n$

TL/L/9992–37

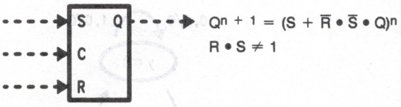
D^n	Q^{n+1}
0	0
1	1



$Q^{n+1} = (\overline{T} \bullet Q + T \bullet \overline{Q})^n$

TL/L/9992–38

T^n	Q^{n+1}
0	Q^n
1	$(\overline{Q})^n$

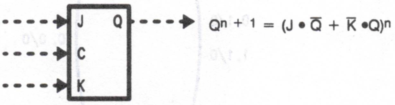


$Q^{n+1} = (S + \overline{R} \bullet \overline{S} \bullet Q)^n$

$R \bullet S \neq 1$

TL/L/9992–39

R	S	Q^{n+1}
0	0	Q^n
0	1	1
1	0	0
1	1	X



$Q^{n+1} = (J \bullet \overline{Q} + \overline{K} \bullet Q)^n$

TL/L/9992–40

J	K	Q^{n+1}
0	0	Q^n
0	1	0
1	0	1
1	1	$(\overline{Q})^n$

FIGURE A.5.1. Basic Flip-Flops

Just as we have a logic gate as the basic combinational circuit element, we have a flip-flop as a basic sequential circuit element. A flip-flop is a memory device which can remember, or store, a binary bit of information. There are four basic flip-flop types: (1) D flip-flop, (2) T flip-flop, (3) RS flip-flop, and (4) JK flip-flop. *Figure A.5.1* shows these elements and their truth table.

With the memory elements, the output does not change as a function of the inputs until the clock transition. Therefore, a superscript notation is used to indicate that the output during clock period n + 1 is a function of the inputs during the previous clock period n.

The D (delay) flip-flop means the input (D) is "stored" in the flip-flop when the clock occurs and will appear on the output (Q) during the next (n + 1) clock time. The D flip-flop is thus very much like a single-bit RAM. It is very useful for data storage and other special applications.

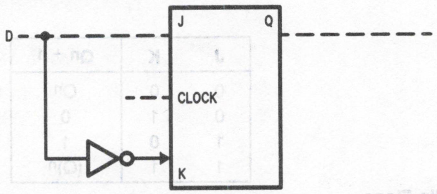
The other three types of flip-flops defined in *Figure A.5.1* are also one-bit storage elements, but instead of simply storing the input, they change state in response to the inputs by various logical rules. Since they hold their previous state in spite of the clock, unless an input goes true, they often simplify the combinational logic functions required to control them in control applications.

The T (toggle) flip-flop, for example, stays in its previous state if the T input is false before the clock. If the T input is true, the output changes to the opposite state (toggle) on the clock. The T flip-flop is thus useful, for example, in binary counters where we want each bit to invert every time there is a carry from the lower order bits.

The R–S flip-flop sets after the S input is true and resets after the R input is true. Its output is undefined if both R and S are true. It is possible to define a Set Overrides Reset (SOR) or a Reset Overrides Set (ROS) flip-flop. It will set or reset respectively if both the R and the S inputs are true.

The J–K flip-flop sets after J is true and resets after K is true. It is similar to an R–S flip-flop except that if J and K are both true, the output changes to the opposite state (toggle). It can be used as a T flip-flop by tying the J and K inputs together.

Since the J–K flip-flop can essentially do the job of both the R–S and the T flip-flop, the R-S and the T flip-flops are seldom seen. The choice is between J–K flip-flops for small counters and control or D flip-flops for data storage applications. Actually the J–K flip-flop can even do the job of the D flip-flop with the addition of a single inverter, as shown in *Figure A.5.2*.



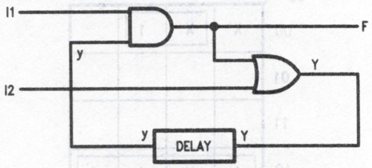
TL/L/9992–41

FIGURE A.5.2. Implement D Flip-Flop by Using J–K

Another memory element type, called a latch, is often described on data sheets with a truth table like the one for the D flip-flop in *Figure A.5.1*. It is definitely not like a D flip-flop, however, because the output changes as soon as the clock goes high and does not "latch" until the clock falls (if the input changes while the clock is high, the output follows it). Because of this characteristic, a latch is not usable in the synchronous logic.

A.6 State Machine Fundamentals

The relationships among present-state variables, primary input variables, next-state (or excitation) variables, and primary output variables that describe the behaviour of a sequential system can be specified in several ways. As an example, consider the simple sequential system that is shown in *Figure A.6.1*.



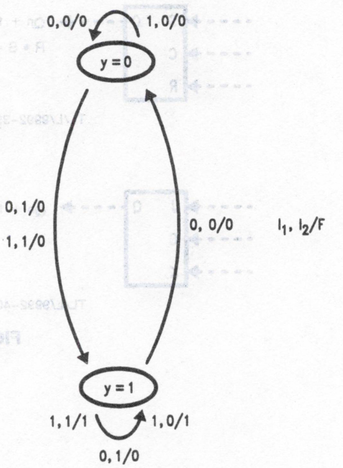
TL/L/9992–42

FIGURE A.6.1. A Typical Sequential Circuit

This system has two primary input variables, having four different combinations of values. There is one primary output variable and one state variable. It uses delay for memory. There are only two possible present states: y = 0 and y = 1. When combined with the four input combinations, these give eight different total present states. The values of the next-state variable, Y, and the primary output variable, F, must be specified for each total present state. The tabular arrangement shown in Table A.6.1 is a common method for presenting this information. This descriptive tool is called a state table.

TABLE A.6.1. State Table

Present State	Next-State Y				Output F			
y	I_1I_2 = 00	01	10	11	I_1I_2 = 00	01	10	11
0	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	1	1



TL/L/9992–43

FIGURE A.6.2. State Diagram

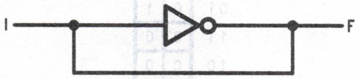
A.6 State Machine Fundamentals (Continued)

A second method for describing the behavior of a sequential system is the use of a state diagram. This method presents a pictorial representation of the present-state/next-state sequences that apply to the sequential device. State changes are marked with directed arrows, with the primary input and output conditions that apply to each state transfer given beside the arrows. The state diagram for the system of *Figure A.6.1* is shown in *Figure A.6.2.* A slash separates the input information from the output information.

State tables and state diagrams are essential tools in the analysis and design of sequential digital systems. The reader should be familiar with these two tools by reading the references listed in the end of this section.

Because a sequential system has feedback from its outputs to its input, certain types of instabilities and uncertainties can occur. When present, these conditions make the operation of circuit difficult or impossible to describe. They may even render the circuit useless, since its behavior may not be predictable or consistent. Several of these types of problems are listed below.

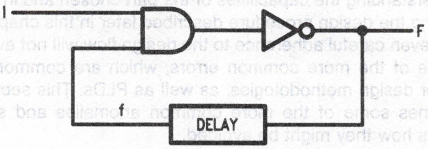
1. The input or output conditions of the system may be indeterminant. For example, the circuit in *Figure A.6.3.*



TL/L/9992-44

FIGURE A.6.3. Example of Hazard Circuit

2. The output condition of the system may be unstable, changing even though the external inputs do not change. *Figure A.6.4* illustrates an example.



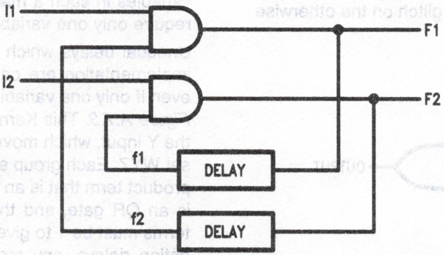
TL/L/9992-45

FIGURE A.6.4. Example of Unstable Circuit

3. The output condition of the system, even though stable, may not be predictable depending upon the primary input conditions. *Figure A.6.5* is an example.

However, these problems mentioned above can be avoided by making certain restrictions in the way sequential systems are designed and used. For instance, the following are some restrictions:

1. Avoiding continuing instabilities (oscillations).
2. Allowing only fundamental-mode operation.
3. Allowing only pulse-mode operation.



TL/L/9992-46

FIGURE A.6.5. Example of Circuit with Unpredictable Output States

A.7 Avoiding Logic Hazards

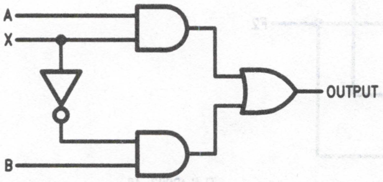
The flexible alternative which PLDs provide to design with standard logic requires care in understanding criteria specific to the new design methodology. Care must be taken in understanding the capabilities of the part chosen and in following the design procedure described later in this chapter. But even careful adherence to the design flow will not avoid some of the more common errors, which are common in other design methodologies, as well as PLDs. This section outlines some of the more common anomalies and suggests how they might be avoided.

HAZARDS AND GLITCHES

Not all devices have the same propagation delay. A hazard may be caused by configuring a set of gates such that a change in the input signals can cause a spurious output signal or "glitch". In combinational circuits, the hazard will be prevented since the outputs are presumed to be a function of steady-state input signals and are not scanned until all transients have stabilized. However, in sequential circuits, particularly where the outputs of such a combinational circuit are used as inputs to a sequential circuit, glitches may occur.

STATIC AND DYNAMIC HAZARDS

Depending on the initial and final value of the output, there can be two classes of hazards. When these values are the same, extraneous output signals result from a *static* hazard. As an example, the circuit shown in *Figure A.7.1* will exhibit an output glitch due to a static hazard when both inputs A and B are high and the control input is changed from high to low. In a perfect world, the output signal would not change, but the propagation delay of the logic gates (in this case the inverter) will cause a momentary low glitch on the otherwise high output, as shown.



TL/L/9992-1

FIGURE A.7.1. Circuit with Static Hazard

If the initial and final states of the output of a circuit are different, then an extraneous output results from a *dynamic* hazard. As an example, this would be characterized by a circuit which moved through an intermediate state before settling in the final configurations, such as a 0-1-0-1 instead of a clean 0-1.

FUNCTION AND LOGIC HAZARDS

The causes of hazards are classed as either function or logic. *Function* hazards exist when logic is specified with a change in more than one input variable possible simultaneously. *Figure A.7.2* shows a truth table which illustrates this. The circuit is intended to move from stable state XYZ = 000 to stable state XYZ = 101. If the input variable X and Z do not change absolutely simultaneously, an output glitch due to a function hazard will occur. Assume both X and Z transition from 0 to 1 at about the same time, but not simultaneously. If X changes before Z, a momentary state of 100 will exist, giving a transient output of 0 until Z changes and the final output stabilizes at 1. If Z changes before X, the inputs are momentarily 001, which gives an output 0, which changes to 1 as X changes.

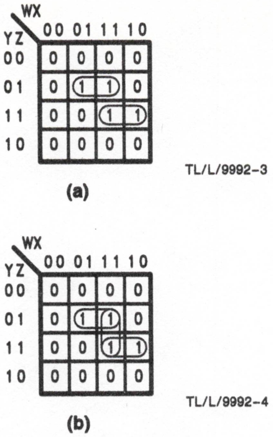
YZ \ X	0	1
00	1	0
01	0	1
11	0	0
10	0	0

TL/L/9992-2

FIGURE A.7.2. Truth Table Illustrating a Function Hazard

Functional glitches can be avoided by assigning the state variables in such a manner that transitions between states require only one variable to change at a time.

Unequal delays which occur because of the detailed logic implementation are called *logic* hazards. These can exist even if only one variable at a time changes, as illustrated by *Figure A.7.3.* This Karnaugh map displays a logic hazard in the Y input, which moves the circuit from the set $X\overline{Y}Z$ to the set WYZ. Each group shown in *Figure A.7.3* represents one product term that is an input to the circuit. In this example, it is an OR gate, and therefore at least one of the product terms must be 1 to give an output of 1. Due to circuit propagation delays, any real-world circuit will move out of the starting sets faster than it moves into the final sets. There is therefore the possibility of a brief interval when neither corresponding product is at 1.



YZ \ WX	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	1	1
10	0	0	0	0

TL/L/9992-3

(a)

YZ \ WX	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	1	1
10	0	0	0	0

TL/L/9992-4

(b)

FIGURE A.7.3 Karnaugh Map (K-Map) Used to Resolve a Function Hazard

A remedy for this is to ensure that any pair between which a transition may take place are in a single set. In other words, any 1-values which appear next to each other in the K-map must be contained within the same set, as shown in *Figure A.7.3.*

REMEDIES FOR MORE COMPLEX CIRCUITS

Once the number of terms exceeds two or three, K-maps become increasingly difficult to work with. A remedy for this can be found by adding additional terms to the original Boolean equations. From this, it can be determined whether a logic hazard exists by examining the modified equations. If a variable and its complement appear in separate product terms in the same equation and these product terms contain that are not mutually exclusive, a logic hazard exists. The hazard can be eliminated by generating a new product term to overlay each pair of product terms which pose a logic hazard. The new product term is selected from canonical product terms which differ only by the state of the variable causing the hazard.

Hazards can exist irrespective of the design methodology used. In manual design, generation and careful examination of K-maps, particularly multiple inputs for state change, can reveal potential hazards. Computer-aided design tools such as ABEL and CUPL are not completely hazard-free and a similar examination of their results may reveal hazards and require adjustment of minimization level and the addition of redundant terms, as for manual design.

As an example of hazard recognition and correction, consider the circuit shown in *Figure A.7.4.* The Boolean equation describing this is:

$$X\overline{Y}Z + WYZ$$

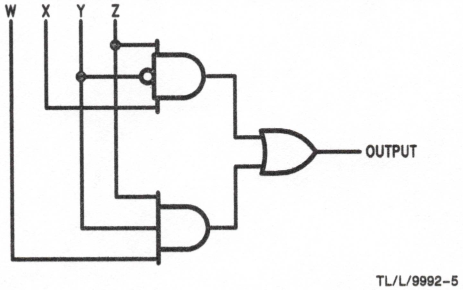
Examining the equation reveals a logic hazard because both Y and $\overline{Y}$ appear in separate product terms and inputs W and X are not mutually exclusive. The problem can be eliminated in two steps. Firstly, expand the expression to its canonical form, which gives:

$$WX\overline{Y}Z + \overline{W}X\overline{Y}Z + WXYZ + W\overline{X}YZ$$

Secondly, develop a new product term from those which overlay the original two and differ only by the state of the variable causing the hazard, in this case Y. This gives:

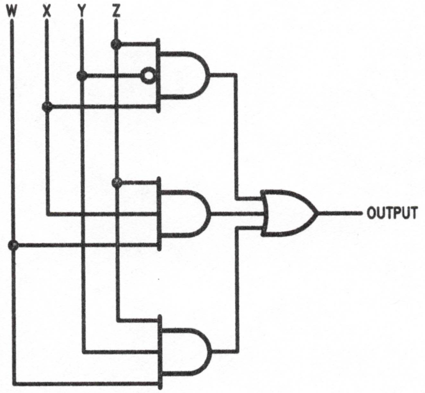
$$X\overline{Y}Z + WXZ\,(Y + \overline{Y}) + WYZ$$
$$= X\overline{Y}Z + WXZ + WYZ$$

In this case, the new product term WXZ overlays the original and is illustrated on the K-map of *Figure A.7.3.* Therefore, the addition of an AND gate and an input to the OR gate will result in elimination of the hazard, as shown in *Figure A.7.4.*



TL/L/9992-5

(a) Logic Hazard Exists



TL/L/9992-6

(b) No Logic Hazard

FIGURE A.7.4. Recognition and Correction of a Logic Hazard

References

Hill & Peterson, "Introduction to Switching Theory and Logical Design".

Kohavi, "Switching and Finite Automata Theory".

Rhyne, "Fundamentals of Digital Systems Designs".

Krieger, "Basic Switching Circuit Theory".

National Semiconductor

Appendix B
Theory of PLD Testing

B.1 Testing Methods

There are many test methods for LSI circuits, each with its own way of generating and processing test data. These approaches can be divided into two broad categories—*concurrent* and *explicit*.[2]

In concurrent approaches, normal user-application input patterns serve as diagnostic patterns. Thus testing and normal computation proceed concurrently. In explicit approaches, on the other hand, special input patterns are applied as tests. Hence, normal computation and testing occur at different times.

CONCURRENT TESTING

Systems that are tested concurrently are designed such that all the information transferred among various parts of the system is coded with different types of error detecting codes. In addition, special circuits monitor this coded data continuously and signal detection of any fault.

Different coding techniques are required to suit the different types of information used inside LSI systems. For example *m*-out-of-*n* codes (*n*-bit patterns with exactly *m* 1's and *n* − *m* 0's) are suitable for coding control signals, while arithmetic codes are best suited for coding ALU operands.[3]

The monitoring circuits—*checkers*—are placed in various locations inside the systems so that they can detect most of the faults. A checker is sometimes designed in a way that enables it to detect a fault in its own circuitry as well as in the monitored data. Such a checker is called a *self-checking checker*.[3]

Hayes and McCluskey surveyed various concurrent testing methods that can be used with microprocessor-based LSI systems.[2] Concurrent testing approaches provide the following advantages:

- Explicit testing expenses (e.g., for test equipment, down time, and test pattern generation) are eliminated during the life of the system, since the data patterns used in normal operation serve as test patterns.
- The faults are detected instantaneously during the use of the LSI chip, hence the first faulty data pattern caused by a certain fault is detected. Thus, the user can rely on the correctness of his output results within the degree of fault coverage provided by the error detection code used. In explicit approaches, on the other hand, nothing can be said about the correctness of the results until the chip is explicitly tested.
- Transient faults, which may occur during normal operation, are detected if they cause any faulty data pattern. These faults cannot be detected by any explicit testing method.

Unfortunately, the concurrent testing approach suffers from several problems that limit its usage in LSI testing:

- The application patterns may not exercise all the storage elements or all the internal connection lines. Defects may exist in places that are not exercised, and hence the faults these defects would produce will not be detected. Thus, the assumption that faults are detected as they occur, or at least before any other fault occurs, is no longer valid. Undetected faults will cause fault accumulation. As a result, the fault detection mechanism may fail because most error detection codes have a limited capability for detecting multiple faults.
- Using error detecting codes to code the information signals used in an LSI chip requires additional I/O pins. At least two extra pins are needed as error signal indicators. (A single pin cannot be used, since such a pin stuck at the good value could go undetected). Because of constraints on pin count, however, such requirements cannot be fulfilled.
- Additional hardware circuitry is required to implement the checkers and to increase the width of the data carriers used for storing and transferring the coded information.
- Designing an LSI circuit for concurrent testing is a much more complicated task than designing a similar LSI circuit that will be tested explicitly.
- Concurrent approaches provide no control over critical voltage or timing parameters. Hence, devices cannot be tested under marginal timing and electrical conditions.
- The degree of fault coverage usually provided by concurrent methods is less than that provided by explicit methods.

The above-mentioned problems have limited the use of concurrent testing for most commercially available LSI circuits. However, as digital systems grow more complex and difficult to test, it becomes increasingly attractive to build test procedures into the UUT (unit under test) itself. We will not consider the concurrent approach further in this article. For a survey of work in concurrent testing, see Hayes and McCluskey.[2]

EXPLICIT TESTING

All explicit testing methods separate the testing process from normal operation. In general, an explicit testing process involves three steps:

- **Generating the test patterns.** The goal of this step is to produce those input patterns which will exercise the UUT under different modes of operation while trying to detect any existing fault.

- **Applying the test patterns to the UUT.** There are two ways to accomplish this step. The first is external testing—the use of special test equipment to apply the test patterns externally. The second is internal testing—the application of test patterns internally by forcing the UUT to execute a self-testing procedure.[2] Obviously, the second method can only be used with systems that can execute programs (for example, with microprocessor-based systems). External testing gives better control over the test process and enables testing under different timing and electrical conditions. On the other hand, internal testing is easier to use because it does not need special test equipment or engineering skills.
- **Evaluating the responses obtained from the UUT.** This step is designed with one of two goals in mind. The first is the detection of an erroneous, which indicates the existence of one or more faults *(go/no-go testing)*. The other is the isolation of the fault, if one exists, in an easily replaceable module *(fault location testing)*. Our interest in this article will be go/no-go testing, since fault location testing of LSI circuits sees only limited use.

Many explicit test methods have evolved in the last decade. They can be distinguished by the techniques used to generate the test patterns and to detect and evaluate the faulty responses *(Figure B.1.1)*. In what follows, we concentrate on explicit testing and present in-depth discussions of the methods of test generation and response evaluation employed with explicit testing.

B.2 Test Generation Techniques

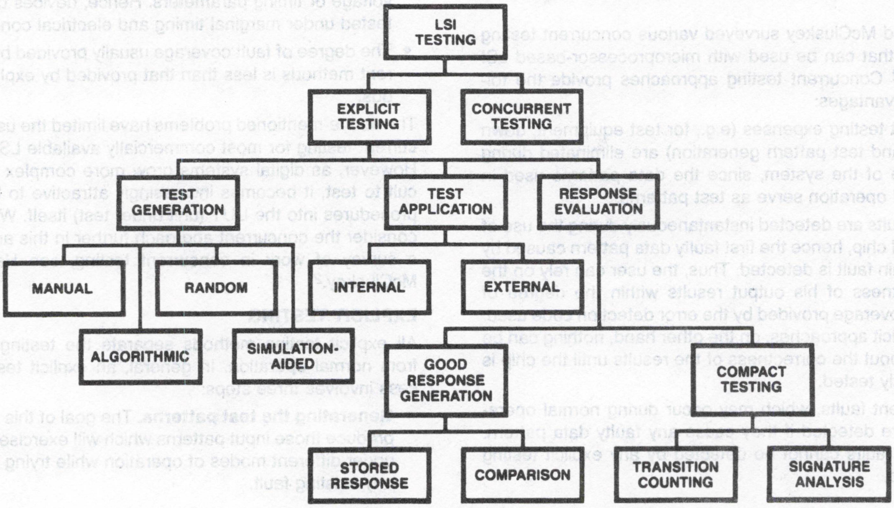
The test generation process represents the most important part of any explicit testing method. Its main goal is to generate those test patterns that, when applied to the UUT, sensitize existing faults and propagate a faulty response to an observable output of the UUT. A test sequence is considered good if it can detect a high percentage of the possible UUT faults; it is considered good, in other words, if its degree of *fault coverage* is high.

Rigorous test generation should consist of three main activities:

- Selecting a good descriptive model, at a suitable level, for the system under consideration. Such a model should reflect the exact behavior of the system in all its possible modes of operation.
- Developing a fault model to define the types of faults that will be considered during test generation. In selecting a fault model, the percentage of possible faults covered by the model should be maximized, and the test costs associated with the use of the model should be minimized. The latter can be accomplished by keeping the complexity of the test generation low and the length of the tests short. Clearly these objectives contradict one another—a good fault model is usually found as a result of a trade-off between them. The nature of the fault model is usually influenced by the model used to describe the system.
- Generating tests to detect all the faults in the fault model. This part of test generation is the soul of the whole test process. Designing a test sequence to detect a certain fault in a digital circuit usually involves two problems. First, the fault must be *excited;* i.e., a certain test sequence must be applied that will force a faulty value to appear at the fault site if the fault exists. Second, the test must be *made sensitive* to the fault; i.e., the effect of the fault must propagate through the network to an observable output.

Rigorous test generation rests heavily on both accurate descriptive (system) models and accurate fault models.

Test generation for digital circuits is usually approached either at the gate-level or at the functional level. The classical approach of modeling digital circuits as a group of connected gates and flip-flops has been used extensively. Using this level of description, test designers introduced many types of fault models, such as the classical stuck-at model. They also assumed that such models could describe physical circuit failures in terms of logic. This assumption has sometimes restricted the number of physical failures that can be modeled, but it has also reduced the complexity of test generation since failures at the elementary level do not have to be considered.



TL/L/9993-1

FIGURE B.1.1. LSI Test Technology

NP-COMPLETE PROBLEMS

The theory of NP-completeness is perhaps the most important theoretical development in algorithm research in the past decade.[29] Its results have meaning for all researchers who are developing computer algorithms.

It is an unexplained phenomenon that for many of the problems we know and study, the best algorithms for their solution have computing times which cluster into two groups. The first group consists of problems whose solution is bounded by a polynomial of small degree. Examples include ordered searching, which is $0(\log n)$, polynomial evaluation, which is $0(n)$, and sorting, which is $0(n \log n)$.[30]

The second group contains problems whose best-known algorithms are nonpolynomial. For example, the best algorithms described in Horowitz and Sahni's book[2] for the traveling salesman and the knapsack problems have a complexity of $0(n^2 2^n)$ and $0(2^{n/2})$, respectively. In the quest to develop efficient algorithms, no one has been able to develop a polynomialtime algorithm for any problem in the second group.

The theory of NP-completeness does not provide a method for obtaining polynomial-time algorithms for these problems. But neither does it say that algorithms of this complexity do not exist. What it does show is that many of the problems for which there is no known polynomial-time algorithm are computationally related. In fact, a problem that is NP-complete has the property that it can be solved in polynomial time if all other NP-complete problems can also be solved in polynomial time.

REFERENCES

29. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1978.

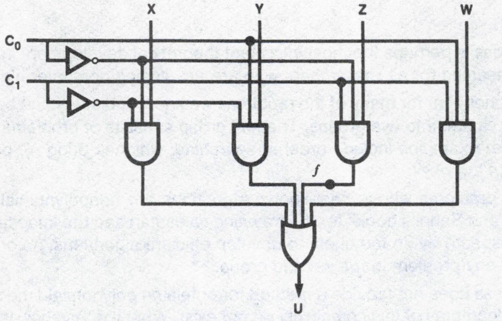
30. E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Washington, D.C., 1978.

Many algorithms have been developed for generating tests for a given fault in combinational networks.(1, 4, 5, 6, 7) However, the complexity of these algorithms depends on the topology of the network; it can become very high for some circuits. Ibarra and Sahni have shown that the problem of generating tests to detect single stuck-at faults in a combinational circuit modeled at the gate level is an NP-complete problem.[8] Moreover, if the circuit is sequential, the problem can become even more difficult depending on the deepness of the circuit's sequential logic.

Thus, for LSI circuits having many thousands of gates, the gate level approach to the test generation problem is not very feasible. A new approach, the functional level, is needed.

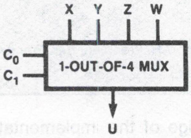
Another important reason for considering faults at the functional level is the constraint imposed on LSI testing by a user environment—the test patterns have to be generated without a knowledge of the implementation details of the chip at the gate level. The only source of information usually available is the typical IC catalog, which details the different modes of operation and describes the general architecture of the circuit. With such information, the test designer finds it easier to define the functional behavior of the circuit and to associate faults with the functions. He can partition the UUT into various modules such as registers, multiplexers, ALUs, ROMs, and RAMs. Each module can be treated as a "black box" performing a specified input/output mapping. These modules can then be tested for *functional failures;* explicit consideration of faults affecting the internal lines is not necessary. The example given below clarifies the idea.

Consider a simple one-out-of-four multiplexers such as the one shown in *Figure B.2.1*. This multiplexer can be modeled at the gate level as shown in *Figure B.2.1(a)*, or at the functional level as shown in *Figure B.2.1.(b)*.



TL/L/9993–2

(a) Gate-Level Description



TL/L/9993–3

C_1	C_0	U
0	0	X
0	1	Y
1	0	Z
1	1	W

(b) Functional-Level Description

FIGURE B.2.1. A One-Out-of-Four Multiplexer

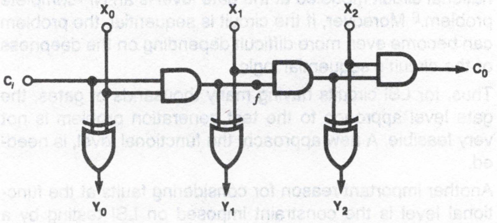
A possible fault model for the gate-level description is the single stuck-at fault model. With this model, the fault list may contain faults such as the line labeled with *f* is stuck at 0, or the control line "C_0" is stuck at 1.

At the functional level, the multiplexer is considered a black box with a well-defined function. Thus, a fault model for it may specify the following as possible faults: selection of wrong source, selection of no source, or presence of stuck-at faults in the input lines or in the multiplexer output. With this model, the fault list may contain faults such as source "X" is selected instead of source "Y", or line "Z" is stuck at 1.

Ad hoc methods—which determine what faults are the most probable—are sometimes used to generate fault lists. But if no fault model is assumed, then the tests derived must be either exhaustive or a rather ad hoc check of the functionality of the system. Exhaustive tests are impossible for even small systems because of the enormous number of possible states, and superficial tests provide neither good coverage nor even an indication of what faults are covered.

Once the fault list has been defined, the next step is to find the test patterns required to detect the faults in the list. As previously mentioned, each fault first has to be excited so that an error signal will be generated somewhere in the UUT. Then this signal has to be sensitized at one of the observable outputs of the UUT. The three examples below describe how to excite and sensitize different types of faults in the types of modules usually encountered in LSI circuits.

Consider the gate-level description of the three-bit incrementer shown in *Figure B.2.2.*



TL/L/9993–4

FIGURE B.2.2. Gate-Level Description of Three-Bit Incrementer

The incrementer output, $Y_2Y_1Y_0$ is the binary sum of C_i and the three-bit binary number $X_2X_1X_0$, while C_0 is the carry-out bit of the sum. Note that $X_0(Y_0)$ is the least significant bit of the incrementer input (output).

Assume we want to detect the fault "line *f* is stuck at 0." To excite that fault we will force a 1 to appear on line *f* so that, if it is stuck at 0, a faulty value will be generated at the fault site. To accomplish this both X_0 and C_i must be set to 1. To sensitize the faulty 0 at *f*, we have to set X_1 to 1; this will propagate the fault to Y_2 independent of the value of X_2. Note that if we set X_1 to 0, the fault will be masked since the AND gate output will be 0, independent of the value at *f*. Note also that X_2 was not specified in the above test. However, by setting X_2 to 1, the fault will propagate to both Y_2 and C_0, which makes the response evaluation task easier.

Consider a microprocessor RAM and assume we want to generate a test sequence to detect the fault "accessing word *i* in the RAM results in accessing the word *j* instead." To excite such a fault, we will use the following sequence of instructions (assume a microprocessor with single-operand instructions):

Load the word 00 . . . 0 into the accumulator.

Store the accumulator contents into memory address *j*.

Load the word 11 . . . 1 into the accumulator.

Store the accumulator contents into memory address *i*.

If the fault exists, these instructions will force a 11 . . . 1 word to be stored in memory address *j* instead of 00 . . . 0. To sensitize the fault, we need only read what is in memory address *j*, using the appropriate instructions. Note that the RAM and its fault have been considered at the functional level, since we did not specify how the RAM is implemented.

Consider the program counter (PC) of a microprocessor and assume we want to generate a test sequence that will detect any fault in the incrementing mode of this PC, i.e., any fault that makes the PC unable to be incremented from x to $x + 1$ for any address x. One way to excite this fault is to force the PC to step through all the possible addresses. This can be easily done by initializing the PC to zero and then executing the no-operation instruction $x + 1$ times. As a result, the PC will contain an address different than $x + 1$. By executing another no-operation instruction, the wrong address can be observed at the address bus and the fault detected. In practice, such an exhaustive test sequence is very expensive, and more economical tests have to be used. Note that, as in the example immediately above, the problem and its solution have been considered at the functional level.

Four methods are currently used to generate test patterns for LSI circuits: manual test generation, algorithmic test generation, simulation-aided test generation, and random test generation.

MANUAL TEST GENERATION

In manual test generation, the test designer carefully analyzes the UUT. This analysis can be done at the gate level, at the functional level or at a combination of the two. The analysis of the different parts of the UUT is intended to determine the specific patterns that will excite and sensitize each fault in the fault list. At one time, the manual approach was widely used for medium- and small-scale digital circuits. Then, the formulation of the D-algorithm and similar algorithms eliminated the need for analyzing each circuit manually and provided an efficient means to generate the required test patterns.[15] However, the arrival of LSI circuits and microprocessors required a shift back toward manual test generation techniques, because most of the algorithmic techniques used with SSI and MSI circuits were not suitable for LSI circuits.

Manual test generation tends to optimize the length of the test patterns and provides a relatively high degree of fault coverage. However, generating tests manually takes a considerable amount of effort and requires persons with special skills. Realizing that test generation has to be done economically, test designers are now moving in the direction of automatic test generation.

One good example of manual test generation is the work done by Sridhar and Hayes,[9] who generated test patterns for a simple bit-sliced microprocessor at the functional level.

A bit-sliced microprocessor is an array of n identical ICs called slices, each of which is a simple processor for operands of k bit length, where k is typically 2 or 4. The interconnections among the n slices are such that the entire array forms a processor for nkbit operands. The simplicity of the individual slices and the regularity of the interconnections make it feasible to use systematic methods for fault analysis and test generation.

Sridhar and Hayes considered a one-bit processor slice as a simplified model for the commercially available bit-sliced processors such as the Am2901.[10] A slice can be modeled as a collection of modules interconnected in a known way. These modules are regarded as black boxes with well-defined input-output relationships. Examples of these functional modules are ALUs, multiplexers, and registers. Combinational modules are described by their truth tables, while sequential modules are defined by their state tables (or state diagrams).

The following fault categories were considered:

- For combinational modules, all possible faults that induce arbitrary changes in the truth table of the module, but that cannot convert it into a sequential circuit.
- For sequential modules, all possible faults that can cause arbitrary changes in the state table of the module without increasing the number of states.

Only one module was assumed to be faulty at any time.

To test for the faults allowed by the above-mentioned fault model, all possible input patterns must be applied to each combinational module (exhaustive testing), and a checking sequence[11] to each sequential module. In addition, the responses of each module must be propagated to observable output lines. The tests required by the individual modules were easily generated manually—a direct consequence of the small operand size ($k = 1$). And because the slices were identical, the tests for one slice were easily extended to the whole array of slices. In fact, Sridhar and Hayes showed that an arbitrary number of simple interconnected slices could be tested with the same number of tests as that required for a single slice, as long as only one slice was faulty at one time. This property is called *C-testability*. Note that the use of carry-lookahead when connecting slices eliminates C-testability. Also note that slices with operand sizes equal to 2 or more usually are not C-testable.

The idea of modeling a digital system as a collection of interconnected functional modules can be used in modeling any LSI circuit. However, using exhaustive tests and checking sequences to test individual modules is feasible only for toy systems. Hence, the fault model proposed by Sridhar and Hayes, though very powerful, is not directly applicable to LSI testing.

PATH SENSITIZATION AND THE D-ALGORITHM

One of the classical fault detection methods at the gate and flip-flop level is the D-algorithm[1, 5] employing the path sensitization testing technique.[4] The basic principle involved in path sensitization is relatively simple. For an input X_i to detect a fault "line a is stuck at j, $j = 0, 1$," the input X_i must cause the signal a in the normal (fault-free) circuit to take the value $\bar{j}$. This condition is necessary but not sufficient to detect the fault. The error signal must be propagated along some path from its site to an observable output.

To generate a test to detect a stuck-at fault in a combinational circuit, the following path sensitization procedure must be followed:

- Excitation—The inputs must be specified so as to generate the appropriate value (0 for stuck-at 1 and 1 for stuck-at 0) at the site of the fault.
- Error propagation—A path from the fault site to an observable output must be selected, and additional signal values to propagate the fault signal along this path must be specified.
- Error propagation—A path from the fault site to an observable output must be selected, and additional signal values to propagate the fault signal along this path must be specified.
- Line justification—Input values must be specified so as to produce the signals values specified in the step above.

There may be several possible choices for error propagation and line justification. Also, in some cases there may be a choice of ways in which to excite the fault. Some of these choices may lead to an inconsistency, and so the procedure must backtrack and consider the next alternative. If all the alternatives lead to an inconsistency, this implies that the fault cannot be detected.

To facilitate the path sensitization process, we introduce the symbol D to represent a signal which has the value 1 in a normal circuit and 0 in a faulty circuit, and $\overline{D}$ to represent a signal which has the value 0 in a normal circuit and 1 in a faulty circuit. The path sensitization procedure can be formulated in terms of a cubical algebra[1, 5] to enable automatic generation of test. This also facilitates test generation for more complex fault models and for fault propagation through complex logic elements.

We shall define three types of cubes (i.e., line values specified in positional notation):

- For a circuit element E which realizes the combinational function f, the "primitive cubes" offer a typical presentation of the prime implicants of f and $\bar{f}$. These cubes concisely represent the logical behavior of E.
- A "primitive D-cube of a fault" in a logic element E specifies the minimal input conditions that must be applied to E in order to produce an error signal (D or $\overline{D}$) at the output of E.
- The "propagation D-cubes" of a logic element E specify the minimal input conditions to the logic element that are required to propagate an error signal on an input (or inputs) to the output of that element.

To generate a test for a stuck-at fault in a combinational circuit, the D-algorithm must perform the following:

1. Fault excitation—A primitive D-cube of the fault under consideration must be selected. This generates the error signal D or $\overline{D}$ at the site of the fault. (Usually a choice exists in this step. The initial choice is arbitrary, and it may be necessary to backtrack and consider another choice).
2. Implication—In Step 1 some of the gate inputs or outputs may be specified so as to uniquely imply values on other signals in the circuit. The implication procedure is performed both forwards and backwards through the circuit. Implication is performed as follows: Whenever a previously unspecified signal value becomes specified, all the elements associated with this signal are placed on a list B and processed one at a time (and removed). For each element processed, it is determined if new values of 0, 1, D, and $\overline{D}$ are implied, based on the previously specified inputs and outputs. These implied line values are determined by intersecting the test cube (which specifies all the previously determined signal values of the circuit) with the primitive cubes of the element. If any line values are implied, the area specified in the test cube, and the associated gates are placed on the list B. An inconsistency occurs when a value is implied on a line which has been specified previously to a different value. If an inconsistency occurs, the procedure must backtrack to the last point a choice existed, reset all lines to their values at that point, and begin again with the next choice.
3. D-propagation—All the elements in the circuit whose output values are unspecified and whose input has some signal D or $\overline{D}$ are placed on a list caled the D-frontier. In this step, an element from the D-frontier is selected and values are assigned to its unspecified inputs so as to propagate the D or $\overline{D}$ on its inputs to one of its outputs. This is accomplished by intersecting the current test cube describing the circuit signal values with a propagation D-cube of the selected element of the D-frontier, resulting in a new test cube. If such intersection is impossible, a new element in the D-frontier is selected. If intersection fails for all the elements in the D-frontier, the procedure backtracks to the last point at which a choice existed.
4. Implication of D-propagation—Implication is performed for the new test cube derived in Step 3.
5. Steps 3 and 4 are repeated until the faulty signal has been propagated to an output of the circuit.

PATH SENSITIZATION AND THE D-ALGORITHM (Continued)

6. Line justification—Execution of Steps 1 to 5 may result in specifying the output value of an element E but leaving some of the inputs to the element unspecified. The unspecified inputs of such an element are assigned values so as to produce the desired output value. This is done by intersecting the test cube with any primitive cube of the element which has no specified signal values that differ from those of the test cube.
7. Implication of line justification—Implication is performed on the new test cube derived in Step 6.
8. Steps 6 and 7 are repeated until all specified element outputs have been justified. Backtracking may again be required.

REFERENCES

1. M.A Breuer and A.D. Friedman, *Diagnosis and Reliable Design of Digital Systems*, Computer Science Press, Washington, D.C., 1976.
4. D.B. Armstrong, "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinatorial Nets," *IEEE Trans. Electronic Computers*, Vol. EC-15, No. 2, Feb. 1966, pp. 63–73.
5. J.P. Roth, W.G. Bouricius, and P.R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," *IEEE Trans. Electronic Computers*, Vol. EC-16, No. 5, Oct. 1967, pp. 567–580.

ALGORITHMIC TEST GENERATION

In algorithmic test generation, the test designer devises a set of algorithms to generate the 1's and 0's needed to test the UUT. Algorithmic test techniques are much more economical than manual techniques. They also provide the test designer with a high level of flexibility. Thus, he can improve the fault coverage of the tests by replacing or modifying parts of the algorithms. Of course, this task is much simpler than modifying the 1's and 0's in a manually generated test sequence.

Techniques that use the gate-level description of the UUT, such as path sensitization[4] and the D-algorithm,[5] can no longer be used in testing complicated LSI circuits. Thus, the problem of generating meaningful sets of tests directly from the functional description of the UUT has become increasingly important. Relatively little work has been done on functional-level testing of LSI chips that are not memory elements.[9,12–17] Functional testing of memory chips is relatively simple because of the regularity of their design and also because their components can be easily controlled and observed from the outside. Various test generation algorithms have been developed to detect different types of faults in memories. [1,18] In the rest of ths section we will concentrate on the general problem of generating tests for irregular LSI chips, i.e., for LSI chips which are not strictly memory chips.

It is highly desirable to find an algorithm that can generate tests for any LSI circuit, or at least most LSI circuits. One good example of work in this area is the technique proposed by Thatte and Abraham for generating tests for microprocessors.[12,13] Another approach, pursued by the authors of this article, is a test generation procedure capable of handling general LSI circuits.[15,16,17]

THE THATTE-ABRAHAM TECHNIQUE

Microprocessors constitute a high percentage of today's LSI circuits. Thatte and Abraham[12,13] approached the microprocessor test generation problem at the functional level.

- The test generation procedure they developed was based on:
- A functional description of the microprocessor at the register-transfer level. The model is defined in terms of data flow among storage units during the execution of an instruction. The functional behavior of a microprocessor is thus described by information about its instruction set and the functions performed by each instruction.
- A fault model describing faults in the various functional parts of the UUT (e.g., the data transfer function, the data storage function, the instruction decoding and control function). This fault model describes the faulty behavior of the UUT without knowing its implementation details.

The microprocessor is modeled by a graph. Each register in the microprocessor (including general-purpose registers and accumulator, stack, program counter, address buffer, and processor status word registers) is represented by a node of the graph. Instructions of the microprocessors are classified as being of transfer, data manipulation, or branch type. There exists a directed edge (labeled with an instruction) from one node to another if during an execution of the instruction data flow occurs from the register represented by the first node to that represented by the second. Examples of instruction representation are given in *Figure B.2.3.*

Having described the function or the structure of the UUT, one needs an appropriate fault model in order to derive useful tests. The approach used by Thatte and Abraham is to partition the various functions of a microprocessor into five classes: the register decoding function, the instruction decoding and control function, the data storage function, the data transfer function, and the data manipulation function. Fault models are derived for each of these functions at a higher level and independently of the details of implementation for the microprocessor. The fault model is quite general. Tests are derived allowing any number of faults, but only in one function at a time; this restriction exists solely to cut down the complexity of test generation.

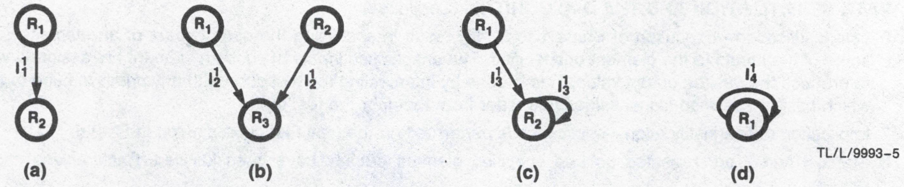


FIGURE B.2.3. Representations of Microprocessor Instruction—I_1,
(a) Transfer Instruction, $R_2 \leftarrow R_1$; (b) Add Instruction, $R_3 \leftarrow R_1 + R_2$;
(c) I_3, OR Instruction, $R_2 \leftarrow R_1$ OR R_2; (d) I_4 Rotate Left Instruction

The fault model for the register decoding function allows any possible set of registers to be accessed instead of a particular register. (If the set is null then no register is accessed.) This fault model is thus very general and independent of the actual realization of the decoding mechanism.

For the instruction decoding and control function, the faulty behavior of the microprocessor is specified as follows. When instruction I_j, is executed any one of the following can happen:

- Instead of instruction I_j, some other instruction I_k is executed. This fault is denoted by $F(I_j/I_k)$.
- In addition to instruction I_j, some other instruction I_k is activated. This fault is denoted by $F(I_j/I_j + I_k)$.
- No instruction is executed. This fault is denoted by $F(I_j/\theta)$.

Under this specification, any number of instructions can be faulty.

In the fault model for the data storage function, any cell in any data storage module is allowed to be stuck at 0 or 1. This can occur in any number of cells.

The fault model for the data transfer function includes the following types of faults:

- A line in a path used in the execution of an instruction is stuck at 0 or 1.
- Two lines of a path used in the instruction are coupled, i.e., they fail to carry different logic values.

Note that the second fault type cannot be modeled by single stuck-at faults. The transfer paths in this fault model are logical paths and thus will account for any failure in the actual physical paths.

Since there is a variety of designs for the ALU and other functional units such as increment or shift logic, no specific fault model is used for the data manipulation function. It is assumed that complete test sets can be derived for the functional units for a given fault model.

By carefully analyzing the logical behavior of the microprocessor according to the fault models presented above, Thatte and Abraham formulated a set of algorithms to generate the necessary test patterns. These algorithms step the microprocessor through a precisely defined set of instructions and addresses. Each algorithm was designed for detecting a particular class of faults, and theorems were proved which showed exactly the kind of faults detected by each algorithm. These algorithms employ the excitation and sensitization concepts previously described.

To gain insight into the problems involved in using the algorithms, Thatte investigated the testing of an eight-bit microprocessor from Hewlett-Packard.[12] He generated the test patterns for the microprocessor by hand, using the algorithms. He found that 96 percent of the single stuck-at faults that could affect the microprocessor were detected by the test sequence he generated. This figure indicates the validity of the technique.

THE ABADIR-REGHBATI TECHNIQUE

Here we will briefly describe a test generation technique we developed for LSI circuits.[15,16] We assume that the tests would be generated in a user environment in which the gate- and flip-flop-level details of the chip were not known. We developed a module-level model for LSI circuits. This model bypasses the gate and flip-flop levels and directly describes blocks of logic (modules) according to their functions. Any LSI circuit can be modeled as a network of interconnected modules such as counters, registers, ALUs, ROMs, RAMs, multiplexers and decoders.

Each module in an LSI circuit was modeled as a black box having a number of functions defined by a set of *binary decision diagrams* (see box).[19] This type of diagram, a functional description tool introduced by Akers in 1978, is a concise means for completely defining the logical operation of one or more digital functions in an implementation-free form. The information usually found in an IC catalog is sufficient to derive the set of binary decision diagrams describing the functions performed by the different modules in a device. These diagrams—like truth tables and state tables—are amenable to extensive logical analysis. However, unlike truth tables and state tables—are amenable to extensive logical analysis. However, unlike truth tables and state tables, they do not have the unpleasant property of growing exponentially with the number of variables involved. Moreover, the diagrams can be stored and processed easily in a digital computer. An important feature of these diagrams is that they state exactly how the module will behave in every one of its operation modes. Such information can be extracted from the module's diagrams in the form of a set of *experiments*.[15,20] Each of these experiments describes the behavior of the module in one of its modes of operation. The structure of these experiments makes them suitable for use in automatic test generation.

We also developed a functional-level fault model describing faulty behavior in the different modules of an LSI chip. This model is quite independent of the details of implementation and covers functional faults that alter the behavior of a module during one of its modes of operation. It also covers stuck-at faults affecting any input or output pin or any interconnection line in the chip.

Using the above-mentioned models, we proposed a functional test generation procedure based on path sensitization and D-algorithm.[15] The procedure takes the module-level model of the LSI chip and the functional description of its modules as parameters and generates tests to detect faults in the fault model. The *fault collapsing technique*[1] was used to reduce the length of the test sequence. As in the D-algorithm, the procedure employs three basic operations, namely implication, D-propagation, and line justification. However, these operations are performed on functional modules.

We also presented algorithmic solutions to the problems of performing these operations on functional modules.[16] For each of the three operations, we gave an algorithm which takes the module's set of experiments and current state (i.e., the values assigned to the module inputs, outputs, and internal memory elements) as parameters and generates all the possible states of the module after performing the required operation.

We have also reported our efforts to develop test sequences based on our test generation procedure for typical LSI circuits.[17] More specifically, we considered a one-bit microprocessor slice C that has all the basic features of the four-bit Am2901 microprocessor slice.[10] The circuit C was modeled as a network of eight functional modules: an ALU, a latch register, an addressable register, and five multiplexers. The functions of the individual modules were described in terms of binary decision diagrams or equivalent sets of experiments. Test capable of detecting various faults covered by the fault model were then generated for the circuit C. We showed that if the fault collapsing technique is used, a significant reduction in the length of the final test sequence results.

The test generation effort was quite straightforward, indicating that the technique can be automated without much difficulty. Our study also shows that for a simplified version of the circuit C the length of the test sequence generated by our technique is very close to the length of the test sequence manually generated by Sridhar and Hayes[9] for the same circuit. We also described techniques for modeling some of the features of the Am2909 four-bit microprogram sequencer[10] that are not covered by the circuit C.

The results of our case study were quite promising and showed that our technique is a viable and effective one for generating tests for LSI circuits.

SIMULATION-AIDED TEST GENERATION

Logic simulation techniques have been used widely in the evaluation and verification of new digital circuits. However; an important application of logic simulation is to interpret the behavior of a circuit under a certain fault or faults. This is known as *fault simulation.* To clarify how this technique can be used to generate tests for LSI systems, we will first describe its use with SSI/MSI-type circuits.

To generate a fault simulator for an SSI/MSI circuit, the following information is needed.[1]

- the gate-level description of the circuit, written in a special language;
- the initial conditions of the memory elements; and
- a list of the faults to be simulated, including classical types of faults such as stuck-at faults and adjacent pin shorts.

The above is fed to a simulation package which generates the fault simulator of the circuit under test. The resulting simulator can simulate the behavior of the circuit under normal conditions as well as when any faults exist.

Now, by applying various input patterns (either generated by hand, by an algorithm, or at random) the simulator checks to see if the output response of the correct circuit differs from one of the responses of the faulty circuits. If it does, then this input pattern detects the fault which created the wrong output response; otherwise the input pattern is useless. If an input pattern is found to detect a certain fault, this fault is deleted from the fault list and the process continues until either the input patterns or the faults are finished. At the end, the faults remaining in the fault list are those which cannot be detected by the input patterns. This direclty measures the degree of fault coverage of the input patterns used.

Two examples of this type of logic simulator are LAMP—the Logic Analyzer for Maintenance Planning developed at Bell Laboratories,[21] and the Testaid III fault simulator developed at the Hewlett-Packard Company.[12] Both work primarily at the gate level and simulate stuck-at faults only. One of the main applications of such fault simulators is to determine the degree of fault coverage provided by a test sequence generated by any other test generation technique.

There are two key requirements that affect the success of any fault simulator:

- the existence of a software model for each primitive element of the circuit, and
- the existence of a good fault model for the UUT which can be used to generate a fault list covering most of the actual physical faults.

These two requirements have been met for SSI/MSI circuits, but they pose serious problems for LSI circuits. If it can be done at all, modeling LSI circuits at the gate level requires great effort. One part of the problem is the lack of detailed information about the internal structure of most LSI chips. The other is the time and memory required to simulate and LSI circuit containing thousands of gates. Another severe problem facing almost all LSI test generation techniques is the lack of good fault models at a level higher than the gate level.

The Abadir-Reghbati description model proposed in the previous section permits the test designer to bypass the gate-level description and, using binary decision diagrams, to define blocks of logic according to their functions. Thus, the simulation of complex LSI circuits can take place at a higher level, and this eliminates the large time and memory requirements. Furthermore, the Abadir-Reghbati fault model is quite efficient and is suitable for simulation purposes. In fact, the implication operation[16] employed by the test generation procedure represents the main building block of any fault simulator. It must be noted that fault simulation techniques are very useful in optimizing the length of the test sequence generated by any test generation technique.

BINARY DECISION DIAGRAMS

Binary decision diagrams are a means of defining the logical operation of digital functions.[19] They tell the user how to determine the output value of a digital function by examining the values of the inputs. Each node in these diagrams is associated with a binary variable, and there are two branches coming out from each node. The right branch is the "1" branch, while the left branch is the "0" branch. Depending on the value of the node variable, one of the two branches will be selected when the diagram is processed.

To see how binary decision diagrams can be used, consider the half-adder shown in *Figure B.2.4(a)*. Assume we are interested in defining a procedure to determine the value of C, given the binary values of X and Y. We can do this by looking at the value of X. If X = 0, then C = 0, and we are finished. If X = 1, we look at Y. If Y = 0, then C = 0, else C = 1, and in either case we are finished. *Figure B.2.4(b)* shows a simple diagram of this procedure. By entering the diagram at the node indicated by the arrow labeled with C and then proceeding through the diagram following the appropriate branches until a 0 or 1 value is reached, we can determine the value C. *Figure B.2.4(c)* shows the diagram representing the function S of the half-adder.

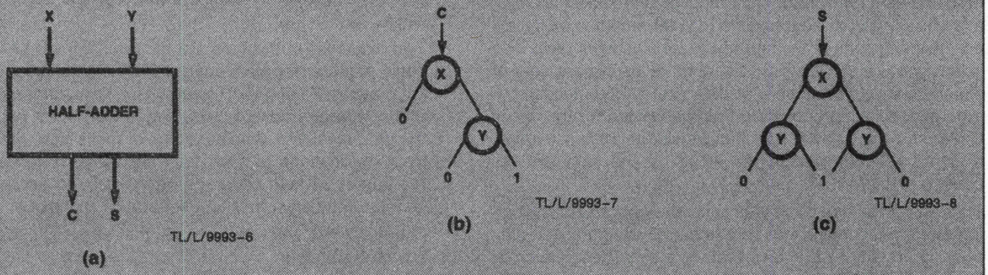


FIGURE B.2.4. (a) Half-Adder; (b) Binary Decision Diagram for $C = X \bullet Y$; (c) Binary Decision Diagram for $S = X \oplus Y$

To simplify the diagrams, any diagram node which has two branches as exit branches can be replaced by the variable itself or its complement. These variables are called exit variables. *Figure B.2.5* shows how this convention is used to simplify the diagrams describing the half-adder.

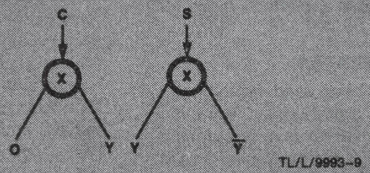


FIGURE B.2.5 Simplified Binary Decision Diagrams for the Half-Adder

In the previous discussion, we have considered only simple diagrams in which the variables within the nodes are primary input variables. However, we can expand the scope of these diagrams by using auxiliary variables as the node variables. These auxiliary variables are defined by their diagrams. Thus, when user encounters such a node variable, say *g*, while tracing a path, he must first process the diagram defining *g* to determine the value of *g*, and then return to the original node and take the appropriate branch. This process is similar to the use of subroutines in high-level programming languages.

For example, consider the full-adder defined by:

$$C_{j+1} = E_j C_j + \overline{E}_j A_j$$

$$S_j = E_j + C_j,$$

where $E_j = A_j + B_j$. *Figure B.2.6* shows the diagrams for these three equations. If the user wants to know the value of C_{j+1} when the values of the three primary inputs A_j, B_j, and C are all 1's, he enters the C_{j+1} diagram, where he encounters

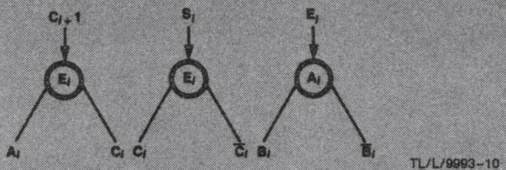


FIGURE B.2.6. Binary Decision Diagrams for a Full-Adder

BINARY DECISION DIAGRAMS (Continued)

the node variable E_j by traversing the E_j diagram, he obtains a value of 0. Returning to the original C_{j+1} diagram with $E_j = 0$ will result in taking the 0 branch and exiting with $C_{j+1} = A_j = 1$.

Since node variables can refer to other auxiliary functions, we can simply describe complex modules by breaking their functions into small subfunctions. Thus, the system diagram will consist of small diagrams connected in a hierarchical structure. Each of these diagrams describes either a module output or an auxiliary variable.

Akers[19] described two procedures to generate the binary decision diagram of a combinational function *f*. The first one uses the truth table description of *f*, while the other uses the boolean expression of *f*. A similar procedure can be derived to generate the binary decision diagram for any sequential function defined by a state table.

Binary decision diagrams can be easily stored and processed by a computer through the use of binary tree structures. Each node can be completely defined by an ordered triple: the node variable and two pointers to the two nodes to which its 0 and 1 branches are directed. Binary decision diagrams can be used in functional testing.[20]

REFERENCES

19. S.B. Akers, "Binary Decision Diagram," *IEEE Trans Computers*, Vol. C-27, No. 6, June 1978, pp. 509–516.

20. S.B. Akers, "Functional Testing with Binary Decision Diagram," *Proc. 8th Int'l Symp. Fault-Tolerant Computing*, June 1978, pp. 82–92.

RANDOM TEST GENERATION

This method can be considered the simplest method for testing a device. A random number generator is used to simultaneously apply random input patterns both to the UUT and to a copy of it known to be fault-free. (This copy is called the *golden unit.*) The results obtained from the two units are compared, and if they do not match, a fault in the UUT is detected. This response evaluation technique is known as comparison testing; we will discuss it later. It is important to note that every time the UUT is tested, a new random test sequence is used.

The important question is how effective the random test is, or, in other words, what fault coverage a random test of given length provides. This question can be answered by employing a fault simulator to simulate the effect of random test patterns of various lengths. The results of such experiments on SSI and MSI circuits show that random test generation is most suitable for circuits without deep sequential logic.[1,22,23] However, by combining random patterns with manually generated ones, test designers can obtain very good results.

The increased sequentiality of LSI circuits reduces the applicability of random testing. Again, combining manually generated test patterns with random ones improves the degree of fault coverage. However, two factors restrict the use of the random test generation technique:

- The dependency on the golden unit, which is assumed to be fault-free, weakens the level of confidence in the results.
- There is no accurate measure of how effective the test is, since all the data gathered about random tests are statistical data. Thus, the amount of fault coverage provided by a particular random test process is unpredictable.

B.3 Response Evaluation Techniques

Different methods have been used to evaluate UUT responses to test patterns. We restrict our discussion to the case where the final goal is only to detect faults or, equivalently, to detect any wrong output response. There are two ways of achieving this goal—using a good response generator or using a compact testing technique.

GOOD RESPONSE GENERATION

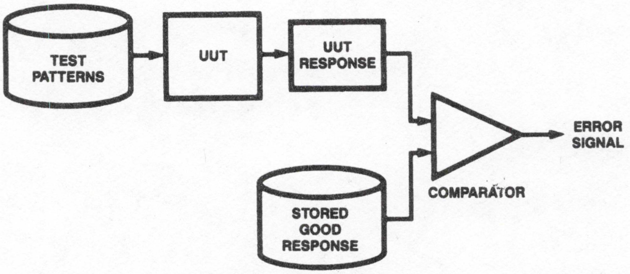
This technique implements an ideal strategy: comparing UUT responses with good response patterns to detect any faulty response. Clearly, the key problems are how to obtain a good response and at what stage in the testing process that response will be generated. In current test systems, two approaches to solving these problems are taken—*stored response testing and comparison testing.*

STORED RESPONSE TESTING

In stored response testing, a one-shot operation generates the good response patterns at the end of the test generation stage. These patterns are stored in an auxiliary memory (usually a ROM). A flow diagram of the stored response testing technique is shown in *Figure B.3.1.*

Different methods can be used to obtain good responses of a circuit to a particular test sequence. One way is to do it manually by analyzing the UUT and the test patterns. This method is the most suitable if the test patterns were generated manually in the first place.

The method most widely used to obtain good responses from the UUT is to apply the test patterns either to a known good copy of the UUT—the golden unit—or to a software-simulated version of the UUT. Of course, if fault simulation techniques were used to generate the test patterns, the UUT's good responses can be obtained very easily as a partial product from the simulator.



TL/L/9993-11

FIGURE B.3.1. Stored Response Testing

The use of a known good device depends on the availability of such a device. Hence, different techniques must be used for the user who wants to test his LSI system and for the designer who wants to test his prototype design. However, golden units are usually available once the device goes into production. Moreover, confidence in the correctness of the responses can be increased by using three or five good devices together to generate the good responses.

The major advantage of the stored response technique is that the good responses are generated only once for each test sequence, thus reducing the cost of the response evaluation step. However, the stored response technique suffers from various disadvantages:

- Any change in the test sequence requires the whole process to be repeated.
- A very large memory is usually needed to store all the good responses to a reasonable test sequence, because both the length and the width of the responses are relatively large. As a result, the cost of testing equipment increases.
- The speed with which the test patterns can be applied to the UUT is limited by the access time of the memory used to store the good responses.

COMPARISON TESTING

Another way to evaluate the responses of the UUT during the testing process is to apply the test patterns simultaneously to both the UUT and a golden unit and to compare their responses to detect any faulty response. The flow diagram of the comparison testing technique is shown in *Figure B.3.2.* The use of comparison testing makes possible the testing of the UUT at different speeds under different electrical parameters, given that these parameters are within the operating limits of the golden unit, which is assumed to be ideal.

Note that in comparison testing the golden unit is used to generate the good responses every time the UUT is tested. In stored response testing, on the other hand, the golden unit is used to generate the good responses only once.

The disadvantages of depending on a golden unit are more serious here, however, since every explicit testing process requires one golden unit. This means that every tester must contain a golden copy of each LSI circuit tested by that tester.

One of the major advantages of comparison testing is that nothing has to be changed in the response evaluation stage if the test sequence is altered This makes comparison testing highly desirable if test patterns are generated randomly.

COMPACT TESTING

The major drawback of good response generation techniques in general, and stored response testing in particular, is the huge amount of response data that must be analyzed and stored. Compact testing methods attempt to solve this by compressing the response data R into a more compact from $f(R)$ from which most of the fault information in R can be derived. Thus, because only the compact form of the good responses has to be stored, the need for large memory or expensive golden units is eliminated. An important property of the compression function f is that it can be implemented with simple circuitry. Thus, compact testing does not require much test equipment and is especially suited for field maintenance work. A general diagram of the compact testing technique is shown in *Figure B.3.3.*

Several choices for the function f exist, such as "the number of 1's in the sequence," "the number of 0 to 1 and 1 to 0 transitions in the sequence" *(transition counting)*,[24] or "the signature of the sequence" *(signature analysis)*.[25] For each compression function f, there is a slight probability that a response R1 different from the fault-free response R0 will be compressed to a form equal to $f(R0)$, i.e., $f(R1) = f(R0)$. Thus, the fault causing the UUT to produce R1 instead of R0 will not be detected, even though it is covered by the test patterns.

The two compression functions that are the most widely accepted commercially are transition counting and signature analysis.

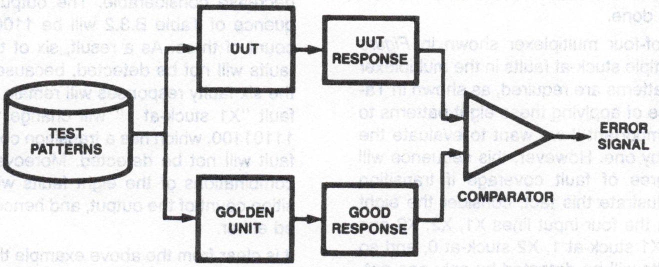


FIGURE B.3.2. Comparison Testing

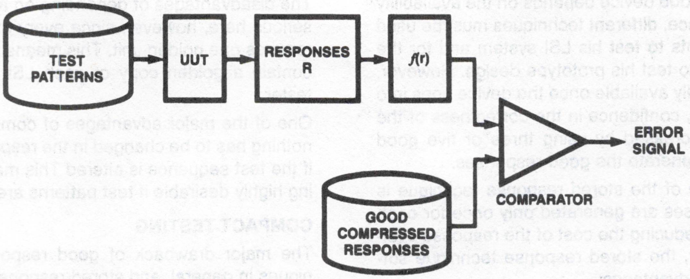
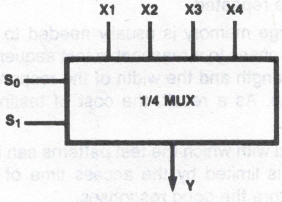


FIGURE B.3.3. Compact Testing



S_0	S_1	Y
0	0	X1
0	1	X2
1	0	X3
1	1	X4

FIGURE B.3.4. One-Out-of-Four Multiplexer

TRANSITION COUNTING

In transition counting, the number of logical transitions (0 to 1 and vice versa) is computed at each output pin by simply running each output of the UUT into a special counter. Thus, the number of counters needed is equal to the number of output pins observed. For every *m*-bit output data stream (at one pin), an *n*-bit counter is required, where $n = [\log_2 m]$. As in stored response testing, the transition counts of the good responses are obtained by applying the test sequence to a golden copy of the UUT and counting the number of transitions at each output pin. This latter information is used as a reference in any explicit testing process.

In the testing of an LSI circuit by means of transition counting, the input patterns can be applied to the UUT at a very high rate, since the response evaluation circuitry is very fast. Also, the size of the memory needed to store the transition counts of the good responses can be very small. For example, a transition counting test using 16 million patterns at a rate of 1 MHz will take 16 seconds, and the compressed stored response will occupy only *K* 24-bit words, where *K* is the number of output pins. This can be contrasted with the 16 million *K*-bit words of storage space needed if regular stored response testing is used.

The test patterns used in a transition counting test system must be designed such that their output responses maximize the fault coverage of the test.[24] The example below shows how this can be done.

Consider the one-out-of-four multiplexer shown in *Figure B.3.4.* To check for multiple stuck-at faults in the multiplexer input lines, eight test patterns are required, as shown in Table B.3.1. The sequence of applying these eight patterns to the multiplexer is not important if we want to evaluate the output responses one by one. However, this sequence will greatly affect the degree of fault coverage if transition counting is used. To illustrate this fact, consider the eight single stuck-at faults in the four input lines X1, X2, X3 and X4 (i.e, X1 stuck-at 0, X1 stuck-at 1, X2 stuck-at 0, and so on). Each of these faults will be detected by only one pattern among the eight test patterns. For example, the fault "X1 stuck-at 0" will be detected by applying the first test pattern in Table B.3.1, but the other seven test patterns will not detect this fault. Now, suppose we want to use transition counting to evaluate the output responses of the multiplexer. Applying the eight test patterns in the sequence shown in Table B.3.1 (from top to bottom) will produce the output response 10101010 (from left to right), with a transition count of seven. Any possible combination of the eight faults described above will change the transition count to a number different from seven, and the fault will be detected. (Note that no more than four of the eight faults can occur at any one time.) Thus, the test sequence shown in Table B.3.1 will detect all single and multiple stuck-at faults in the four input lines of the multiplexers.

Now, if we change the sequence of the test patterns to the one shown in Table B.3.2., the fault coverage of the test will decrease considerable. The output responses of the sequence of Table B.3.2 will be 11001100, with a transition count of three. As a result, six of the eight single stuck-at faults will not be detected, because the transition count of the six faulty responses will remain three. For example, the fault "X1 stuck-at 1" will change the output response to 11101100, which has a transition count of three. Hence, this fault will not be detected. Moreover, most of the multiple combinations of the eight faults will not change the transition count of the output, and hence they will not be detected either.

It is clear from the above example that the order of applying the test patterns to the UUT greatly affects the fault coverage of the test. When testing combinational circuits, the test designer is completely free to choose the order of test patterns. However, he cannot do the same with test patterns for sequential circuits. More seriously, because he is dealing with LSI circuits that probably have multiple output lines, he will find that a particular test sequence may give good results at some outputs and bad results at others. One way to solve these contradictions is to use simulation techniques to find the optimal test sequence. However, because of the limitations discussed here, transition counting cannot be recognized as a powerful compact LSI testing method.

TABLE B.3.1. The Eight Test Patterns Used for Testing the Multiplexer of *Figure B.3.4*

S_0	S_1	X1	X2	X3	X4	Y
0	0	1	0	0	0	1
0	0	0	1	1	1	0
0	1	0	1	0	0	1
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	1	1	1	0	0

TABLE B.3.2. A Different Sequence of the Eight Multiplexer Test Patterns

S_0	S_1	X1	X2	X3	X4	Y
0	0	1	0	0	0	1
0	1	0	1	0	0	1
0	0	0	1	1	1	0
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	1	0	0	0	1	1
1	0	1	1	0	1	0
1	1	1	1	1	0	0

SIGNATURE ANALYSIS

In 1977 Hewlett-Packard Corporation introduced a new compact testing technique called signature analysis, intended for testing LSI systems.$^{25-28}$ In this method, each output response is passed through a 16-bit linear feedback shift register whose contents $f(R)$, after the test patterns have been applied, are called the test *signature*. *Figure B.3.5* shows an example of a linear feedback shift register used in signature analysis.

The signature provided by linear feedback shift registers can be regarded as a unique fingerprint—hence, test designers have extremely high confidence in these shift registers as tools for catching errors. To better understand this confidence, let us examine the 16-bit linear feedback shift register shown in *Figure B.3.5*. Let us assume a data stream of length n is fed to the serial data input line (representing the output response to be evaluated). There are 2^n possible combinations of data streams, and each one will be compressed to one of the 2^{16} possible signatures. Linear feedback shift registers have the property of equally distributing the different combinations of data streams over the different signatures.[27] This property is illustrated by the following numerical examples.

- Assume $n = 16$. Then each data stream will be mapped to a distinctive signature (one-to-one mapping).
- Assume $n = 17$. Then exactly two data streams will be mapped to the same signature. Thus, for a particular data stream (the UUT good output response), there is only one other data stream (a faulty output response) that will have the same signature; i.e., only one fault response out of $2^{17}-1$ possible faults will not be detected.
- Assume $n = 18$. Then four different data streams will be mapped to the same signature. Hence, only three faults out of $2^{18} - 1$ possible faults will not be detected.

We can generalize the results obtained above. For any response data stream of length $n > 16$, the probability of missing a faulty response when using a 16-bit signature analyzer is[27]

$$\frac{2^{n-16}-1}{2^n-1} \approx 2^{-16}, \text{ for } n >> 16.$$

Hence, the possibility of missing an error in the bit stream is very small (on the order of 0.002 percent). Note also that a great percentage of the faults will affect more than one output pin—hence the probability of not detecting these kind of faults is even lower.

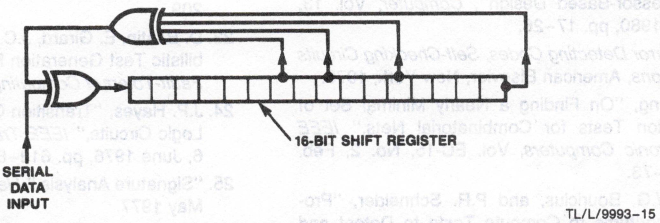


FIGURE B.3.5. The 16-Bit Linear Feedback Shift Register Used in Signature Analysis

Signature analysis provides a much higher level of confidence for detecting faulty output responses than that provided by transition counting. But, like transition counting, it requires only very simple hardware circuitry and a small amount of memory for storing the good signatures. As a result, the signatures of the output responses can be calculated even when the UUT is tested at its maximum speed. Unlike transition counting, the degree of fault coverage provided by signature analysis is not sensitive to the order of the test patterns. Thus, it is clear that signature analysis is the most attractive solution to the response evaluation problem.

The rapid growth of the complexity and performance of digital circuits presents a testing problem of increasing severity. Although many testing methods have worked well for SSI and MSI circuits, most of them are rapidly becoming obsolete. New techniques are required to cope with the vastly more complicated LSI circuits.

In general, testing techniques fall into the concurrent and explicit categories. In this article, we gave special attention to explicit testing techniques, especially those approaching the problem at the functional level. The explicit testing process can be partitioned into three steps: generating the test, applying the test to the UUT, and evaluating the UUT's responses. The various testing techniques are distinguished by the methods they used to perform these three steps. Each of these techniques has certain strengths and weaknesses.

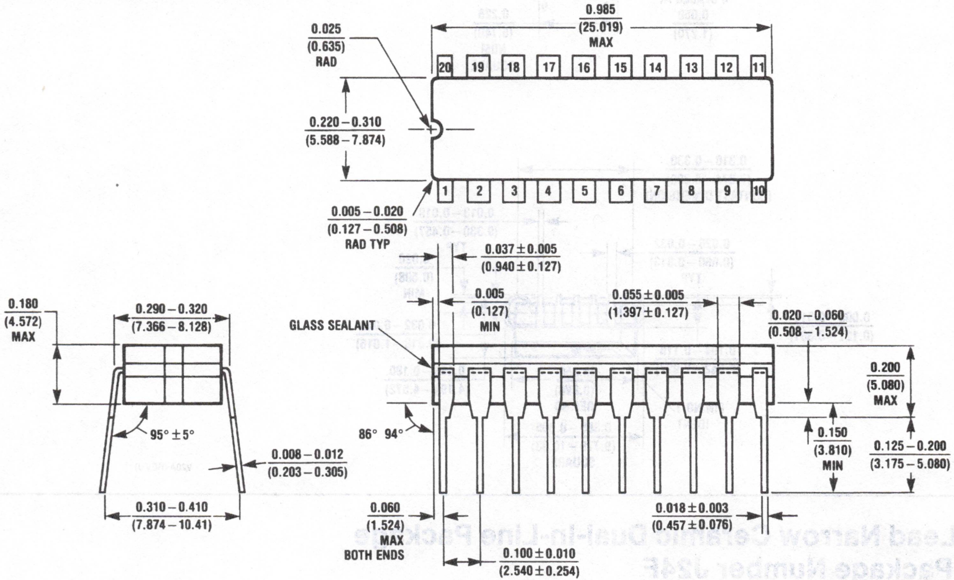
We have tried to emphasize the range of testing techniques available, and to highlight some of the milestones in the evolution of LSI testing. The details of an individual test method can be found in the source we have cited.

References

1. M.A. Breuer and A.D. Friedman, *Diagnosis and Reliable Design of Digital Systems*, Computer Science Press, Washington, D.C., 1976.
2. J.P. Hayes and E.J. McCluskey, "Testing Considerations in Microprocessor-Based Design", *Computer*, Vol. 13, No.3, March 1980, pp. 17–26.
3. J. Wakerly, *Error Detecting Codes, Self-Checking Circuits and Applications*, American Elsevier, New York, 1978.
4. D.B. Armstrong, "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinatorial Nets," *IEEE Trans. Electronic Computers*, Vol. EC-15, No. 2, Feb. 1966, pp. 63–73.
5. J.P. Roth, W.G. Bouricius, and P.R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," *IEEE Trans. Electronica Computers*, Vol. EC-16, No. 5, October 1967, pp. 567–580.
6. S.B. Akers, "Test Generation Techniques," *Computer*, Vol. 13, No. 3, March 1980, pp. 9–15.
7. E.I. Muehldorf and A.D. Savkar, "LSI Logic Testing—An Overview," *IEEE Trans. Computers*, Vol. C-30, No. 1, January 1981, pp. 1–17.
8. O.H. Ibarra and S.K. Sahni, "Polynomially Complete Fault Detection Problems," *IEEE Trans. Computers*, Vol. C-24, No. 3, March 1975, pp. 242–249.
9. T. Sridhar and J.P. Hayes, "Testing Bit-Sliced Microprocessors," *Proc. 9th Int'l Symp. Fault-Tolerant Computing*, 1979, pp. 211–218.
10. *The Am2900 Family Data Book*, Advanced Micro Devices, Inc., 1979.
11. Z. Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1970.
12. S.M. Thatte, "Test Generation for Microprocessors," PhD Thesis, University of Illinois, Urbana, 1979.
13. S.M. Thatte and J.A. Abraham, "Test Generation For Microprocessors," *IEEE Trans. Computers*, Vol. C-29, No. 6, June 1980, pp. 429–441.
14. M.A. Breuer and A.D. Friedman, "Functional Level Primitives in Test Generation," *IEEE Trans. Computers*, Vol. C-29, No. 3, March 1980, pp. 223–235.
15. M.S. Abadir and H.K. Reghbati, "Test Generation for LSI: A New Approach," Tech. Report 81-7, Dept. of Computational Science, University of Saskatchewan, Saskatoon, 1981.
16. M.S. Abadir and H.K. Reghbati, "Test Generation for LSI: Basic Operations," Tech. Report 81-8, Dept. of Computational Science, University of Saskatchewan, Saskatoon, 1981.
17. M.S. Abadir and H.K. Reghbati, "Test Generation for LSI: A Case Study," Tech. Report 81-9, Dept. of Computational Science, University of Saskatchewan, Saskatoon, 1981.
18. M.S. Abadir and H.K. Reghbati, "Functional Testing of Semiconductor Random Access Memories," Tech. Report 81-6, Dept. of Computational Science, University of Saskatchewan, Saskatoon, 1981.
19. S.B. Akers, "Binary Decision Diagram," *IEEE Trans. Computers*, Vol. C-27, No. 6, June 1978, pp. 509–516.
20. S.B. Akers, "Functional Testing with Binary Decision Diagram," *Proc. 8th Int'l Symp. Fault-Tolerant Computing*, June 1978, pp. 82–92.
21. B.A Zimmer, "Test Techniques for Circuit Boards Containing Large Memories and Microprocessors," *Proc. 1976 Semiconductor Test Symp.*, pp. 16–21.
22. P. Agrawal and V.D. Agrawal, "On Improving the Efficiency of Monte Carlo Test Generation," *Proc. 5th Int'l Symp. Fault-Tolerant Computing*, June 1975, pp. 205–209.
23. D. Bastin, E. Girard, J.C. Rault, and R. Tulloue, "Probabilistic Test Generation Methods," *Proc. 3rd Int'l Symp. Fault-Tolerant Computing*, June 1973, p. 171.
24. J.P. Hayes, "Transition Count Testing of Combinational Logic Circuits," *IEEE Trans. Computers*, Vol. C-25, No. 6, June 1976, pp. 613–620.
25. "Signature Analysis," *Hewlett Packard J.*, Vol. 28, No. 9, May 1977.
26. R. David, "Feedback Shift Register Testing," *Proc. 8th Int'l Symp. Fault-Tolerant Computing*, June 1978.
27. H.J. Nadig, "Testing a Microprocessor Product Using Signature Analysis," *Proc. 1978 Semiconductor Test Symp.*, pp. 159–169.
28. J.B. Peatman, *Digital Hardware Design*, McGraw-Hill, New York, 1980.
29. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1978.
30. E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Washington D.C., 1978.

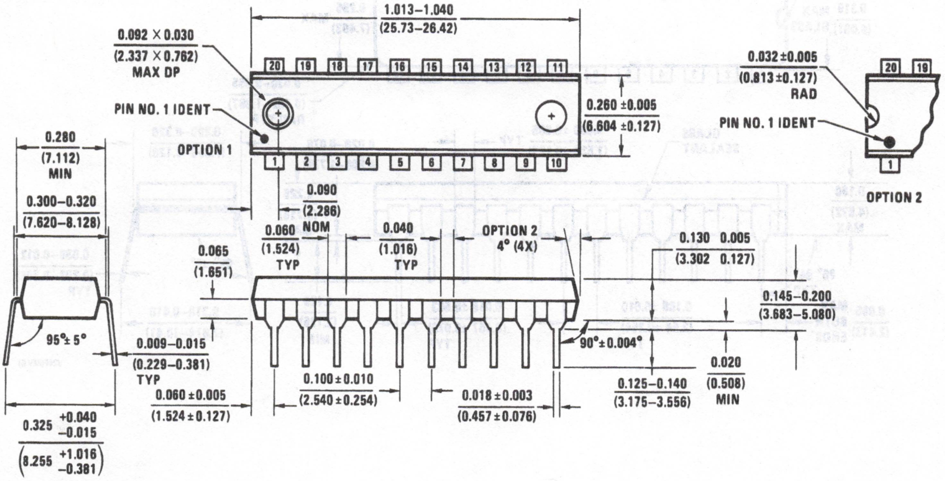
Appendix C
Physical Dimensions

20-Lead Ceramic Dual-In-Line Package
NS Package Number J20A


0.985
(25.019)
MAX
0.025
(0.635)
RAD
20
19
18
17
16
15
14
13
12
11
0.220–0.310
(5.588–7.874)
1
2
3
4
5
6
7
8
9
10
0.005–0.020
(0.127–0.508)
RAD TYP
0.037 ± 0.005
(0.940 ± 0.127)
0.005
(0.127)
MIN
0.055 ± 0.005
(1.397 ± 0.127)
0.020–0.060
(0.508–1.524)
0.180
(4.572)
MAX
0.290–0.320
(7.366–8.128)
GLASS SEALANT
0.200
(5.080)
MAX
95° ± 5°
86° 94°
0.150
(3.810)
MIN
0.125–0.200
(3.175–5.080)
0.008–0.012
(0.203–0.305)
0.310–0.410
(7.874–10.41)
0.060
(1.524)
MAX
BOTH ENDS
0.018 ± 0.003
(0.457 ± 0.076)
0.100 ± 0.010
(2.540 ± 0.254)

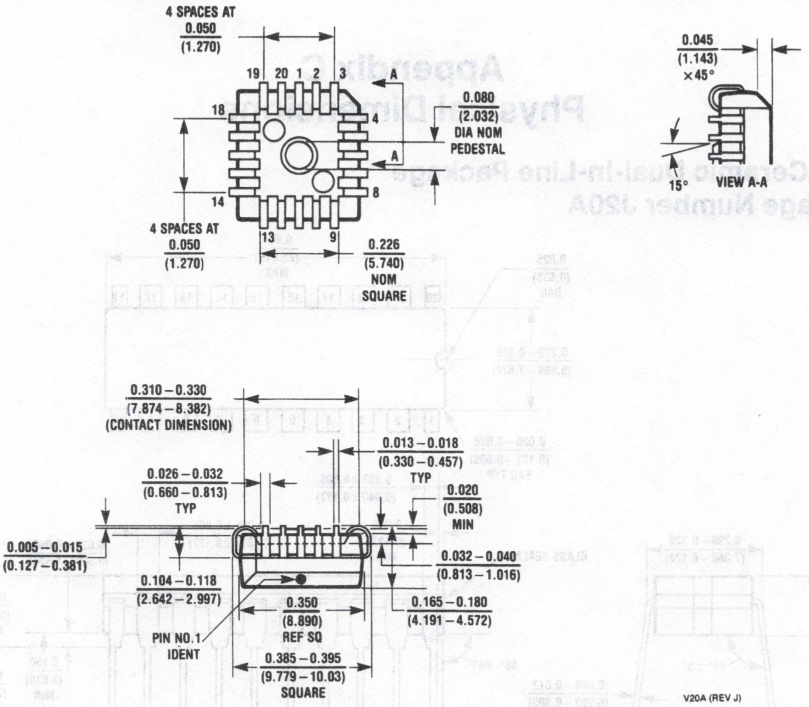
J20A (REV M)

20-Lead Dual-In-Line Package
NS Package Number N20A


1.013–1.040
(25.73–26.42)
0.092 × 0.030
(2.337 × 0.762)
MAX DP
20
19
18
17
16
15
14
13
12
11
PIN NO. 1 IDENT
OPTION 1
0.260 ± 0.005
(6.604 ± 0.127)
1
2
3
4
5
6
7
8
9
10
0.032 ± 0.005
(0.813 ± 0.127)
RAD
PIN NO. 1 IDENT
OPTION 2
0.280
(7.112)
MIN
0.300–0.320
(7.620–8.128)
0.090
(2.286)
NOM
0.060
(1.524)
TYP
0.040
(1.016)
TYP
OPTION 2
4° (4X)
0.130 0.005
(3.302 0.127)
0.065
(1.651)
0.145–0.200
(3.683–5.080)
95° ± 5°
0.009–0.015
(0.229–0.381)
TYP
90° ± 0.004°
0.020
(0.508)
MIN
0.060 ± 0.005
(1.524 ± 0.127)
0.100 ± 0.010
(2.540 ± 0.254)
0.018 ± 0.003
(0.457 ± 0.076)
0.125–0.140
(3.175–3.556)
0.325 +0.040 −0.015
(8.255 +1.016 −0.381)

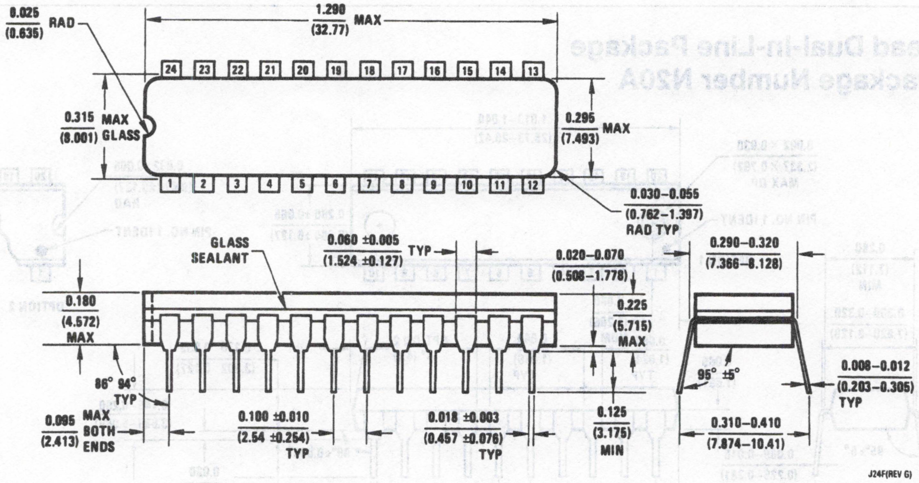
N20A (REV G)

20-Lead Plastic Chip Carrier
NS Package Number V20A


4 SPACES AT
0.050
(1.270)
19 20 1 2 3
A
18
4
0.080
(2.032)
DIA NOM
PEDESTAL
A
8
14
13 9
4 SPACES AT
0.050
(1.270)
0.226
(5.740)
NOM
SQUARE
0.045
(1.143)
×45°
15°
VIEW A-A
0.310 – 0.330
(7.874 – 8.382)
(CONTACT DIMENSION)
0.013 – 0.018
(0.330 – 0.457)
TYP
0.026 – 0.032
(0.660 – 0.813)
TYP
0.020
(0.508)
MIN
0.005 – 0.015
(0.127 – 0.381)
0.032 – 0.040
(0.813 – 1.016)
0.104 – 0.118
(2.642 – 2.997)
0.350
(8.890)
REF SQ
0.165 – 0.180
(4.191 – 4.572)
PIN NO.1
IDENT
0.385 – 0.395
(9.779 – 10.03)
SQUARE

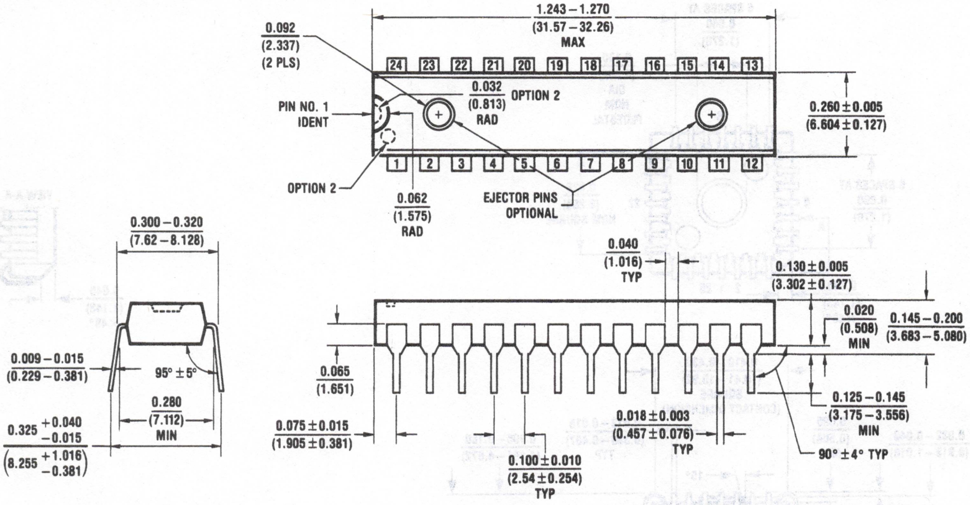
V20A (REV J)

24-Lead Narrow Ceramic Dual-In-Line Package
NS Package Number J24F


0.025
(0.635)
RAD
1.290
(32.77)
MAX
24 23 22 21 20 19 18 17 16 15 14 13
0.315 MAX
(8.001) GLASS
0.295
(7.493)
MAX
1 2 3 4 5 6 7 8 9 10 11 12
0.030–0.055
(0.762–1.397)
RAD TYP
GLASS
SEALANT
0.060 ±0.005
(1.524 ±0.127)
TYP
0.020–0.070
(0.508–1.778)
0.290–0.320
(7.366–8.128)
0.180
(4.572)
MAX
0.225
(5.715)
MAX
95° ±5°
0.008–0.012
(0.203–0.305)
TYP
86° 94°
TYP
0.095
(2.413)
MAX
BOTH
ENDS
0.100 ±0.010
(2.54 ±0.254)
TYP
0.018 ±0.003
(0.457 ±0.076)
TYP
0.125
(3.175)
MIN
0.310–0.410
(7.874–10.41)

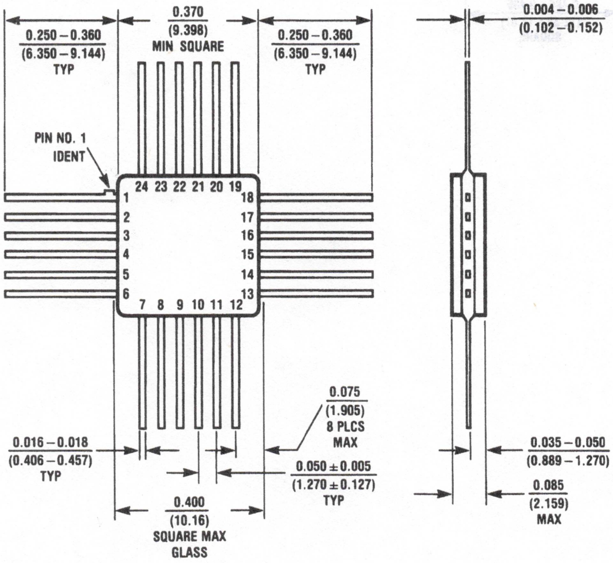
J24F(REV G)

24-Lead Narrow Plastic Dual-In-Line Package
NS Package Number N24C


1.243 – 1.270
(31.57 – 32.26)
MAX
0.092
(2.337)
(2 PLS)
24 23 22 21 20 19 18 17 16 15 14 13
0.032
(0.813)
RAD
OPTION 2
PIN NO. 1
IDENT
0.260 ± 0.005
(6.604 ± 0.127)
OPTION 2
1 2 3 4 5 6 7 8 9 10 11 12
0.062
(1.575)
RAD
EJECTOR PINS
OPTIONAL
0.300 – 0.320
(7.62 – 8.128)
0.040
(1.016)
TYP
0.130 ± 0.005
(3.302 ± 0.127)
0.020
(0.508)
MIN
0.145 – 0.200
(3.683 – 5.080)
0.009 – 0.015
(0.229 – 0.381)
95° ± 5°
0.065
(1.651)
0.280
(7.112)
MIN
0.125 – 0.145
(3.175 – 3.556)
MIN
0.325 +0.040 −0.015
(8.255 +1.016 −0.381)
0.075 ± 0.015
(1.905 ± 0.381)
0.018 ± 0.003
(0.457 ± 0.076)
TYP
90° ± 4° TYP
0.100 ± 0.010
(2.54 ± 0.254)
TYP

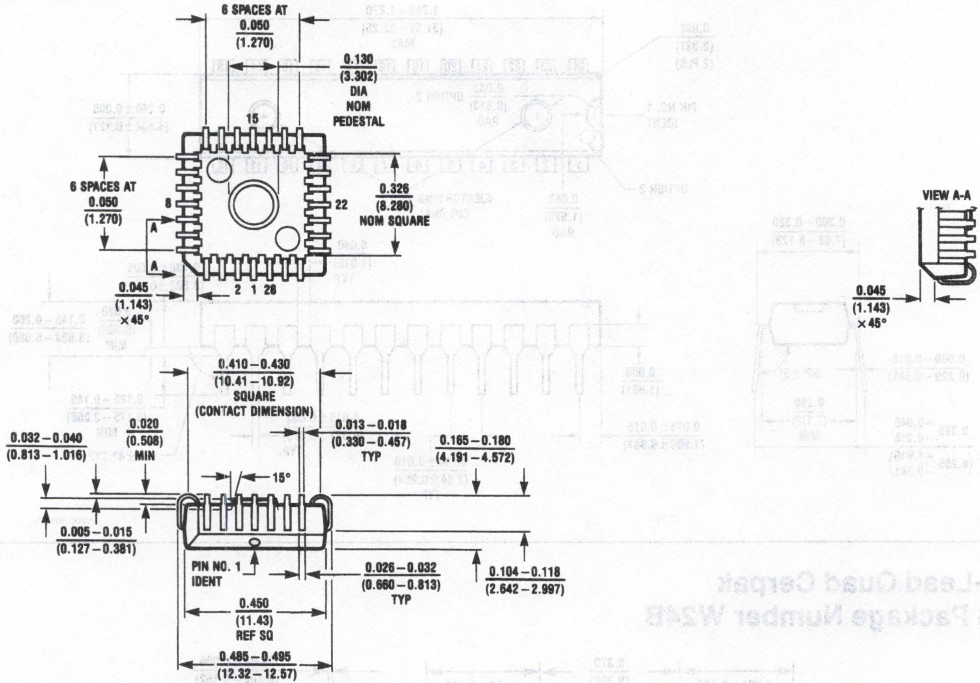
N24C (REV F)

24-Lead Quad Cerpak
NS Package Number W24B


0.250 – 0.360
(6.350 – 9.144)
TYP
0.370
(9.398)
MIN SQUARE
0.250 – 0.360
(6.350 – 9.144)
TYP
0.004 – 0.006
(0.102 – 0.152)
PIN NO. 1
IDENT
24 23 22 21 20 19
1 2 3 4 5 6
18 17 16 15 14 13
7 8 9 10 11 12
0.075
(1.905)
8 PLCS
MAX
0.016 – 0.018
(0.406 – 0.457)
TYP
0.050 ± 0.005
(1.270 ± 0.127)
TYP
0.035 – 0.050
(0.889 – 1.270)
0.085
(2.159)
MAX
0.400
(10.16)
SQUARE MAX
GLASS

W24B (REV C)

28-Lead Plastic Chip Carrier
NS Package Number V28A


6 SPACES AT
0.050
(1.270)
0.130
(3.302)
DIA
NOM
PEDESTAL
15
6 SPACES AT
0.050
(1.270)
8
22
A
A
0.326
(8.280)
NOM SQUARE
0.045
(1.143)
× 45°
2 1 28
0.410 – 0.430
(10.41 – 10.92)
SQUARE
(CONTACT DIMENSION)
0.020
(0.508)
MIN
0.032 – 0.040
(0.813 – 1.016)
0.013 – 0.018
(0.330 – 0.457)
TYP
0.165 – 0.180
(4.191 – 4.572)
15°
0.005 – 0.015
(0.127 – 0.381)
PIN NO. 1
IDENT
0.026 – 0.032
(0.660 – 0.813)
TYP
0.104 – 0.118
(2.642 – 2.997)
0.450
(11.43)
REF SQ
0.485 – 0.495
(12.32 – 12.57)
SQUARE
VIEW A-A
0.045
(1.143)
× 45°

V28A (REV G)

NOTES

NOTES

NOTES

NOTES

National Semiconductor

Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you informed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement.

Please send them to:

Technical Communications Dept. M/S 23-200
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

For a recorded update of this listing plus ordering information for these books from National's Literature Distribution operation, please call (408) 749-7378.

ALS/AS LOGIC DATABOOK—1987

Introduction to Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky

ASIC DESIGN MANUAL/GATE ARRAYS & STANDARD CELLS—1987

SSI/MSI Functions • Peripheral Functions • LSI/VLSI Functions • Design Guidelines • Packaging

ASIC 1.5 MICRON CMOS GATE ARRAY—1988

Library of Macros • Functional Blocks • Packaging • Design Flow • Workstation Integration
Military Screening Program (MSP) • Design Guidelines

CMOS LOGIC DATABOOK—1988

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC
MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • Surface Mount

DATA COMMUNICATION/LAN/UART DATABOOK—Rev. 1—1988

LAN IEEE 802.3 • High Speed Serial/IBM Data Communications • ISDN Components • UARTs
Modems • Transmission Line Drivers/Receivers

DRAM MANAGEMENT HANDBOOK—1988

Dynamic Memory Control • Error Detection and Correction • Microprocessor Applications for the DP8408A/09A/17/18/19/28/29 • Microprocessor Applications for the DP8420A/21A/22A

F100K DATABOOK—1989

Family Overview • F100K Datasheets • 11C Datasheets • 10K and 100K Memory Datasheets
Design Guide • Circuit Basics • Logic Design • Transmission Line Concepts • System Considerations
Power Distribution and Thermal Considerations • Testing Techniques • Quality Assurance and Reliability

FACT™ ADVANCED CMOS LOGIC DATABOOK

Reprint of Fairchild 1987 Databook

Description and Family Characteristics • Ratings, Specifications and Waveforms
Design Considerations • 54AC/74ACXXX • 54ACT/74ACTXXX

FAST® ADVANCED SCHOTTKY TTL LOGIC DATABOOK—1988

Circuit Characteristics • Ratings, Specifications and Waveforms • Design Considerations • 54F/74FXXX

GRAPHICS DATABOOK—1988

Advanced Graphics Chipset • Application Notes

INTERFACE DATABOOK—1988

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral Power Drivers • Display Drivers
Memory Support • Microprocessor Support • Level Translators and Buffers • Frequency Synthesis • Hi-Rel Interface

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LINEAR 1 DATABOOK—1988

Voltage Regulators • Operational Amplifiers • Buffers • Voltage Comparators • Instrumentation Amplifiers • Surface Mount

LINEAR 2 DATABOOK—1988

Active Filters • Analog Switches/Multiplexers • Analog-to-Digital • Digital-to-Analog • Sample and Hold
Sensors • Voltage References • Surface Mount

LINEAR 3 DATABOOK—1988

Audio Circuits • Radio Circuits • Video Circuits • Motion Control • Special Functions • Surface Mount

LS/S/TTL DATABOOK—1987

Introduction to Bipolar Logic • Low Power Schottky • Schottky • TTL • Low Power

MASS STORAGE HANDBOOK—1988

Winchester Disk Preamplifiers • Winchester Disk Servo Control • Winchester Disk Pulse Detectors
Winchester Disk Data Separators/Synchronizers and ENDECs • Winchester Disk Data Controller
SCSI Bus Interface Circuits • Floppy Disk Controllers

MEMORY DATABOOK—1988

PROMs, EPROMs, EEPROMs • Flash EPROMs and EEPROMs • TTL I/O SRAMs
ECL I/O SRAMs • ECL I/O Memory Modules

MICROCONTROLLER DATABOOK—1988

COP400 Family • COP800 Family • COPS Applications • HPC Family • HPC Applications
MICROWIRE and MICROWIRE/PLUS Peripherals • Display/Terminal Management Processor (TMP)
Microcontroller Development Tools

SERIES 32000 MICROPROCESSORS DATABOOK—1988

Series 32000 Overview • Central Processing Units • Slave Processors • Peripherals • Board Level Products
Development Systems and Tools • Software Support • Application Notes • NSC800 Family

RELIABILITY HANDBOOK—1986

Reliability and the Die • Internal Construction • Finished Package • MIL-STD-883 • MIL-M-38510
The Specification Development Process • Reliability and the Hybrid Device • VLSI/VHSIC Devices
Radiation Environment • Electrostatic Discharge • Discrete Device • Standardization
Quality Assurance and Reliability Engineering • Reliability and Documentation • Commercial Grade Device
European Reliability Programs • Reliability and the Cost of Semiconductor Ownership
Reliability Testing at National Semiconductor • The Total Military/Aerospace Standardization Program
883B/RETS™ Products • MILS/RETS™ Products • 883/RETS™ Hybrids • MIL-M-38510 Class B Products
Radiation Hardened Technology • Wafer Fabrication • Semiconductor Assembly and Packaging
Semiconductor Packages • Glossary of Terms • Key Government Agencies • AN/ Numbers and Acronyms
Bibliography • MIL-M-38510 and DESC Drawing Cross Listing

TELECOMMUNICATIONS—1987

Line Card Components • Integrated Services Digital Network Components • Modems
Analog Telephone Components • Application Notes

NATIONAL SEMICONDUCTOR CORPORATION DISTRIBUTORS

ALABAMA
Huntsville
Arrow Electronics
(205) 837-6955
Bell Industries
(205) 837-1074
Hamilton/Avnet
(205) 837-7210
Pioneer
(205) 837-9300

ARIZONA
Phoenix
Arrow Electronics
(602) 437-0750
Tempe
Anthem Electronics
(602) 966-6600
Bell Industries
(602) 966-7800
Hamilton/Avnet
(602) 961-6400

CALIFORNIA
Agora Hills
Zeus Components
(818) 889-3838
Anaheim
Time Electronics
(714) 934-0911
Zeus Components
(714) 921-9000
Chatsworth
Anthem Electronics
(818) 700-1000
Arrow Electronics
(818) 701-7500
Hamilton Electro Sales
(818) 700-6500
Time Electronics
(818) 998-7200
Costa Mesa
Avnet Electronics
(714) 754-6050
Hamilton Electro Sales
(714) 641-4159
Garden Grove
Bell Industries
(714) 895-7801
Gardena
Bell Industries
(213) 515-1800
Hamilton Electro Sales
(213) 217-6751
Irvine
Anthem Electronics
(714) 768-4444
Ontario
Hamilton/Avnet
(714) 989-4602
Rocklin
Bell Industries
(916) 969-3100
Roseville
Bell Industries
(916) 969-3100
Sacramento
Anthem Electronics
(916) 922-6800
Hamilton/Avnet
(916) 925-2216
San Diego
Anthem Electronics
(619) 453-9005
Arrow Electronics
(619) 565-4800
Hamilton/Avnet
(619) 571-7510
Time Electronics
(619) 586-1331
San Jose
Anthem Electronics
(408) 295-4200
Zeus Components
(408) 998-5121
Sunnyvale
Arrow Electronics
(408) 745-6600
Bell Industries
(408) 734-8570
Hamilton/Avnet
(408) 743-3355
Time Electronics
(408) 734-9888
Thousand Oaks
Bell Industries
(805) 499-6821
Torrance
Time Electronics
(213) 320-0880
Tustin
Arrow Electronics
(714) 838-5422
Yorba Linda
Zeus Components
(714) 921-9000

COLORADO
Englewood
Anthem Electronics
(303) 790-4500
Arrow Electronics
(303) 790-4444
Hamilton/Avnet
(303) 799-9998
Wheatridge
Bell Industries
(303) 424-1985

CONNECTICUT
Cheshire
Time Electronics
(203) 271-3200
Danbury
Hamilton/Avnet
(203) 797-2800
Meridan
Anthem Electronics
(203) 237-2282
Norwalk
Pioneer Northeast
(203) 853-1515
Wallingford
Arrow Electronics
(203) 265-7741

FLORIDA
Altamonte Springs
Arrow/Kierulff Electronics
(305) 682-6923
Pioneer
(305) 834-9090
Deerfield Beach
Arrow Electronics
(305) 429-8200
Bell Industries
(305) 421-1997
Pioneer
(305) 428-8877
Fort Lauderdale
Hamilton/Avnet
(305) 971-2900
Lake Mary
Arrow Electronics
(407) 323-0252
Largo
Bell Industries
(813) 541-4434
Orlando
Chip Supply
(305) 298-7100
Oviedo
Zeus Components
(407) 365-3000
Palm Bay
Arrow Electronics
(305) 725-1480
St. Petersburg
Hamilton/Avnet
(813) 576-3930
Winter Park
Hamilton/Avnet
(407) 628-3888

GEORGIA
Norcross
Arrow Electronics
(404) 449-8252
Bell Industries
(404) 662-0923
Hamilton/Avnet
(404) 447-7500
Pioneer
(404) 448-1711

ILLINOIS
Bensenville
Hamilton/Avnet
(312) 860-7780
Elk Grove Village
Anthem Electronics
(312) 640-6066
Bell Industries
(312) 640-1910
Itasca
Arrow Electronics
(312) 250-0500
Urbana
Bell Industries
(217) 328-1077

INDIANA
Carmel
Hamilton/Avnet
(317) 844-9333
Fort Wayne
Bell Industries
(219) 423-3422
Indianapolis
Advent Electronics Inc.
(317) 872-4910
Arrow Electronics
(317) 243-9353
Bell Industries
(317) 634-8202
Pioneer
(317) 849-7300

IOWA
Cedar Rapids
Advent Electronics
(319) 363-0221
Arrow Electronics
(319) 395-7230
Bell Industries
(319) 395-0730
Hamilton/Avnet
(319) 362-4757

KANSAS
Lenexa
Arrow Electronics
(913) 541-9542
Overland Park
Hamilton/Avnet
(913) 888-8900

MARYLAND
Columbia
Anthem Electronics
(301) 995-6640
Arrow Electronics
(301) 995-0003
Hamilton/Avnet
(301) 995-3500
Lionex Corp.
(301) 964-0040
Time Electronics
(301) 964-3090
Zeus Components
(301) 997-1118
Gaithersburg
Pioneer
(301) 921-0660

MASSACHUSETTS
Lexington
Pioneer Northeast
(617) 861-9200
Zeus Components
(617) 863-8800
Norwood
Gerber Electronics
(617) 769-6000
Peabody
Hamilton/Avnet
(617) 531-7430
Sertech Laboratories
(617) 532-5105
Time Electronics
(617) 532-6200
Wilmington
Anthem Electronics
(617) 657-5170
Arrow Electronics
(617) 935-5134
Lionex Corporation
(617) 657-5170

MICHIGAN
Ann Arbor
Arrow Electronics
(313) 971-8220
Bell Industries
(313) 971-9093
Grand Rapids
Arrow Electronics
(616) 243-0912
Hamilton/Avnet
(616) 243-8805
Pioneer Standard
(616) 698-1800
Livonia
Hamilton/Avnet
(313) 522-4700
Pioneer
(313) 525-1800
Wyoming
R. M. Michigan, Inc.
(616) 531-9300

MINNESOTA
Eden Prairie
Anthem Electronics
(612) 944-5454
Pioneer-Twin Cities
(612) 944-3355
Edina
Arrow Electronics
(612) 830-1800
Minnetonka
Hamilton/Avnet
(612) 932-0600

MISSOURI
Earth City
Hamilton/Avnet
(314) 344-1200
St. Louis
Arrow Electronics
(314) 567-6888
Time Electronics
(314) 391-6444

NEW HAMPSHIRE
Hudson
Bell Industries
(603) 882-1133
Manchester
Arrow Electronics
(603) 668-6968
Hamilton/Avnet
(603) 624-9400

NATIONAL SEMICONDUCTOR CORPORATION DISTRIBUTORS (Continued)

NEW JERSEY
Cherry Hill
Hamilton/Avnet
(609) 424-0100
Fairfield
Hamilton/Avnet
(201) 575-3390
Lionex Corporation
(201) 227-7960
Marlton
Arrow Electronics
(609) 596-8000
Parsippany
Arrow Electronics
(201) 538-0900
Pine Brook
Nu Horizons Electronics
(201) 882-8300
Pioneer
(201) 575-3510

NEW MEXICO
Albuquerque
Alliance Electronics Inc.
(505) 292-3360
Arrow Electronics
(505) 243-4566
Bell Industries
(505) 292-2700
Hamilton/Avnet
(505) 765-1500

NEW YORK
Amityville
Nu Horizons Electronics
(516) 226-6000
Binghamton
Pioneer
(607) 722-9300
Buffalo
Summit Distributors
(716) 887-2800
Fairport
Pioneer Northeast
(716) 381-7070
Hauppauge
Anthem Electronics
(516) 273-1660
Arrow Electronics
(516) 231-1000
Hamilton/Avnet
(516) 434-7413
Lionex Corporation
(516) 273-1660
Time Electronics
(516) 273-0100
Port Chester
Zeus Components
(919) 937-7400
Rochester
Arrow Electronics
(716) 427-0300
Hamilton/Avnet
(716) 475-9130
Summit Electronics
(716) 334-8100
Ronkonkoma
Zeus Components
(516) 737-4500
Syracuse
Hamilton/Avnet
(315) 437-2641
Time Electronics
(315) 432-0355
Westbury
Hamilton/Avnet
(516) 997-6868

NORTH CAROLINA
Charlotte
Pioneer
(704) 527-8188
Durham
Pioneer Technology
(919) 544-5400
Raleigh
Arrow Electronics
(919) 876-3132
Hamilton/Avnet
(919) 878-0810
Winston-Salem
Arrow Electronics
(919) 725-8711

OHIO
Centerville
Arrow Electronics
(513) 435-5563
Cleveland
Pioneer
(216) 587-3600
Dayton
Bell Industries
(513) 435-8660
Bell Industries
(513) 434-8231
Hamilton/Avnet
(513) 439-6700
Pioneer
(513) 236-9900
Zeus Components
(914) 937-7400
Highland Heights
CAM/Ohio Electronics
(216) 461-4700
Solon
Arrow Electronics
(216) 248-3990
Hamilton/Avnet
(216) 831-3500
Westerville
Hamilton/Avnet
(614) 882-7004

OKLAHOMA
Tulsa
Arrow Electronics
(918) 252-7537
Hamilton/Avnet
(918) 252-7297
Quality Components
(918) 664-8812
Radio Inc.
(918) 587-9123

OREGON
Beaverton
Almac-Stroum Electronics
(503) 629-8090
Anthem Electronics
(503) 643-1114
Arrow Electronics
(503) 645-6456
Lake Oswego
Bell Industries
(503) 241-4115
Hamilton/Avnet
(503) 635-7850

PENNSYLVANIA
Horsham
Anthem Electronics
(215) 443-5150
Lionex Corp.
(215) 443-5150
Pioneer
(215) 674-4000
King of Prussia
Time Electronics
(215) 337-0900
Monroeville
Arrow Electronics
(412) 856-7000
Pittsburgh
Hamilton/Avnet
(412) 281-4150
Pioneer
(412) 782-2300
CAM/RPC Ind. Elec.
(412) 782-3770

TEXAS
Addison
Quality Components
(214) 733-4300
Austin
Arrow Electronics
(512) 835-4180
Hamilton/Avnet
(512) 837-8911
Pioneer
(512) 835-4000
Quality Components
(512) 835-0220
Minco Technology Labs
(512) 834-2022
Carrollton
Arrow Electronics
(214) 380-6464
Dallas
Pioneer
(214) 386-7300
Houston
Arrow Electronics
(713) 530-4700
Pioneer
(713) 988-5555
Irving
Hamilton/Avnet
(214) 550-7755
Richardson
Anthem Electronics
(214) 238-0237
Zeus Components
(214) 783-7010
Stafford
Hamilton/Avnet
(713) 240-7733
Sugarland
Quality Components
(713) 240-2255

UTAH
Midvale
Bell Industries
(801) 972-6969
Salt Lake City
Anthem Electronics
(801) 973-8555
Arrow Electronics
(801) 973-6913
Bell Industries
(801) 972-6969
Hamilton/Avnet
(801) 972-4300

WASHINGTON
Bellevue
Almac-Stroum Electronics
(206) 643-9992
Hamilton/Avnet
(206) 453-5844
Kent
Arrow Electronics
(206) 575-4420
Redmond
Anthem Electronics
(206) 881-0850
Hamilton/Avnet
(206) 867-0148

WISCONSIN
Brookfield
Arrow Electronics
(414) 792-0150
Mequon
Taylor Electric
(414) 241-4321
Waukesha
Bell Industries
(414) 547-8879
Hamilton/Avnet
(414) 784-4516

CANADA
WESTERN PROVINCES
Burnaby
Hamilton/Avnet
(604) 437-6667
Semad Electronics
(604) 438-2515
Calgary
Hamilton/Avnet
(403) 250-9380
Semad Electronics
(403) 252-5664
Zentronics
(403) 272-1021
Edmonton
Zentronics
(403) 468-9306
Richmond
Zentronics
(604) 273-5575
Saskatoon
Zentronics
(306) 955-2207
Winnipeg
Zentronics
(204) 694-1957

EASTERN PROVINCES
Brampton
Zentronics
(416) 451-9600
Mississauga
Hamilton/Avnet
(416) 677-7432
Nepean
Hamilton/Avnet
(613) 226-1700
Zentronics
(613) 226-8840
Ottawa
Semad Electronics
(613) 727-8325
Pointe Claire
Semad Electronics
(514) 694-0860
St. Laurent
Hamilton/Avnet
(514) 335-1000
Zentronics
(514) 737-9700
Waterloo
Zentronics
(800) 387-2329
Willowdale
ElectroSound Inc.
(416) 494-1666

SALES OFFICES

ALABAMA
Huntsville
(205) 721-9367

ARIZONA
Tempe
(602) 966-4563

CALIFORNIA
Inglewood
(213) 645-4226
Roseville
(916) 786-5577
San Diego
(619) 587-0666
Santa Clara
(408) 562-5900
Tustin
(714) 259-8880
Woodland Hills
(818) 888-2602

COLORADO
Boulder
(303) 440-3400
Colorado Springs
(303) 578-3319
Englewood
(303) 790-8090

CONNECTICUT
Hamden
(203) 288-1560

FLORIDA
Boca Raton
(305) 997-8133
Orlando
(305) 629-1720
St. Petersburg
(813) 577-1380

GEORGIA
Norcross
(404) 441-2740

ILLINOIS
Schaumburg
(312) 397-8777

INDIANA
Carmel
(317) 843-7160
Fort Wayne
(219) 484-0722

IOWA
Cedar Rapids
(319) 395-0090

KANSAS
Overland Park
(913) 451-8376

MARYLAND
Hanover
(301) 796-8900

MASSACHUSETTS
Burlington
(617) 273-3170

MICHIGAN
Grand Rapids
(616) 940-0588
W. Bloomfield
(313) 855-0166

MINNESOTA
Bloomington
(612) 854-8200

NEW JERSEY
Paramus
(201) 599-0955

NEW MEXICO
Albuquerque
(505) 884-5601

NEW YORK
Fairport
(716) 223-7700
Melville
(516) 351-1000
Wappinger Falls
(914) 298-0680

NORTH CAROLINA
Cary
(919) 481-4311

OHIO
Dayton
(513) 435-6886
Highland Heights
(216) 461-0191

ONTARIO
Mississauga
(416) 678-2920
Nepean
(613) 596-0411

OREGON
Portland
(503) 639-5442

PENNSYLVANIA
Horsham
(215) 675-6111

PUERTO RICO
Rio Piedias
(809) 758-9211

QUEBEC
Lachine
(514) 636-8525

TEXAS
Austin
(512) 346-3990
Houston
(713) 771-3547
Richardson
(214) 234-3811

UTAH
Salt Lake City
(801) 322-4747

WASHINGTON
Bellevue
(206) 453-9944

WISCONSIN
Brookfield
(414) 782-1818

National Semiconductor

National Semiconductor
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

SALES OFFICES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA
Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F. Mexico
Tel: 52-5-524-9402

National Semicondutores
Do Brasil Ltda.
Av. Brig. Faria Lima, 1383
6.0 Andor-Conj. 62
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Fax: (55/11) 211-1181 NSBR BR

National Semiconductor GmbH
Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: 49-08141-103-0
Telex: 527 649

National Semiconductor (UK) Ltd.
301 Harpur Centre
Horne Lane
Bedford MK40 ITR
United Kingdom
Tel: (02 34) 27 00 27
Telex: 826 209

National Semiconductor Benelux
Vorstlaan 100
B-1170 Brussels
Belgium
Tel: (02) 6725360
Telex: 61007

National Semiconductor (UK) Ltd.
1, Bianco Lunos Alle
DK-1868 Fredriksberg C
Denmark
Tel: (01) 213211
Telex: 15179

National Semiconductor
Expansion 10000
28, rue de la Redoute
F-92260 Fontenay-aux-Roses
France
Tel: (01) 46 60 81 40
Telex: 250956

National Semiconductor S.p.A.
Strada 7, Palazzo R/3
20089 Rozzano
Milanofiori
Italy
Tel: (02) 8242046/7/8/9

National Semiconductor AB
Box 2016
Stensatravagen 13
S-12702 Skarholmen
Sweden
Tel: (08) 970190
Telex: 10731

National Semiconductor
Calle Agustin de Foxa, 27
28036 Madrid
Spain
Tel: (01) 733-2958
Telex: 46133

National Semiconductor
Switzerland
Alte Winterthurerstrasse 53
Postfach 567
Ch-8304 Wallisellen-Zurich
Switzerland
Tel: (01) 830-2727
Telex: 59000

National Semiconductor
Kauppakartanonkatu 7
SF-00930 Helsinki
Finland
Tel: (0) 33 80 33
Telex: 126116

National Semiconductor Japan
Ltd.
Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-ku
Tokyo 160 Japan
Tel: 3-299-7001
Fax: 3-299-7000

National Semiconductor
Hong Kong Ltd.
Southeast Asia Marketing
Austin Tower, 4th Floor
22-26A Austin Avenue
Tsimshatsui, Kowloon, H.K.
Tel: 852 3-7243645
Cable: NSSEAMKTG
Telex: 52996 NSSEA HX

National Semiconductor
(Australia) PTY, Ltd.
1st Floor, 441 St. Kilda Rd.
Melbourne, 3004
Victory, Australia
Tel: (03) 267-5000
Fax: 61-3-2677458

National Semiconductor (PTE),
Ltd.
200 Cantonment Road 13-01
Southpoint
Singapore 0208
Tel: 2252226
Telex: RS 33877

National Semiconductor (Far East)
Ltd.
Taiwan Branch
P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (86) 02-501-7227
Telex: 22837 NSTW
Cable: NSTW TAIPEI

National Semiconductor (Far East)
Ltd.
Korea Office
Room 612,
Korea Fed. of Small Bus. Bldg.
16-2, Yoido-Dong,
Youngdeungpo-Ku
Seoul, Korea
Tel: (02) 784-8051/3 – 785-0696-8
Telex: K24942 NSRKLO

© 1989 National Semiconductor
TL/2467P RRD-RRD85M019/Printed in U.S.A.